(NASA-CR-4202) A COMPARISON OF NUMERICAL METHODS FOR THE PREDICTION OF TWO-DIMENSIONAL HEAT TRANSFER IN AN ELECTROTHERMAL DEICER PAD M.S. Thesis. Final Contractor Report (Toledo Univ.)

N89-13429

Unclas

H1/06 0179683

NASA Contractor Report 4202

A Comparison of Numerical Methods for the Prediction of Two-Dimensional Heat Transfer in an Electrothermal Deicer Pad

William B. Wright
University of Toledo
Toledo, Ohio

Prepared for
Lewis Research Center
under Grant NAG3-72

National Aeronautics
and Space Administration

Scientific and Technical
Information Division

1988

ACKNOWLEDGEMENTS

I would like to thank Drs. Kenneth J. De Witt and Theo G. Keith for their guidance in the completion of this thesis, and for acting as my advisors. Dr. Millard L. Jones deserves thanks for serving on my committee with Dr. De Witt and Dr. Keith. Dr. Stephen E. LeBlanc deserves special thanks for allowing me use of the Macintosh in his laboratory, on which this thesis was written.

Appreciation is also acknowledged for financial support provided by the Department of Chemical Engineering, the University of Toledo, and NASA Lewis Research Center, Cleveland, Ohio, during the period of this work.

PRECEDING PAGE BLANK NOT FILMED

PAGE ii INTENTIONALLY BLANK

TABLE OF CONTENTS

ACKNOWLEGEMENTS iii

NOMENCLATURE vii

ABSTRACT x

I. INTRODUCTION 1

II. LITERATURE REVIEW 3

III. EXTENSION OF TWO-DIMENSIONAL DEICER PROGRAM 7

IV. FORMULATION OF THE NUMERICAL METHOD 10

- A. Governing Equations and Boundary Conditions 10
- B. Finite Differencing Procedures 13
  - 1. Simple Explicit 14
  - 2. Crank-Nicolson 16
  - 3. Simple Implicit 18
  - 4. Hopscotch 19
  - 5. Alternating Direction Explicit 20
  - 6. Alternating Direction Implicit 23
  - 7. Time-Splitting 26
- C. Application of Boundary Conditions 27

V. NUMERICAL SOLUTION METHODS 48

- A. Gauss-Seidel Iteration 48
- B. SIP and MSIP Methods 50
- C. Method of Assumed States 52

VI. DISCUSSION OF RESULTS 54

- A. Accuracy of Methods 54
- B. Program Efficiency 58

PRECEDING PAGE BLANK NOT FILMED

PAGE iv INTENTIONALLY BLANK

C. Parameter Variation 60
  1. Gap Size 60
  2. Grid Spacing 68
  3. Ice Thickness 68
D. Comparison with Experimental Data 76
VII. CONCLUSIONS AND RECOMMENDATIONS 87
VIII. REFERENCES 89
APPENDIX: Flow Diagram, Computer Listing and Sample Input Data 92

NOMENCLATURE

C_p	specific heat capacity (BTU/lb-^oF)
H	enthalpy (BTU/lb)
h	heat transfer coefficient (BTU/hr-ft^2-^oF)
k	thermal conductivity (BTU/hr-ft-^oF)
L_f	latent heat of fusion for ice (BTU/lb)
m	number of nodes in y-direction of the grid
n	number of nodes in x-direction of the grid
q	rate of heat generation per unit area (BTU/hr-ft^2 or W/in^2)
T	temperature (^oF)
t	time variable (hr)
x	space coordinate in x-direction (ft)
y	space coordinate in y-direction (ft)

Greek Letters:

α	thermal diffusivity (ft^2/hr)
Δt	size of time step (hr)
Δx	size of grid spacing in x-direction (ft)
Δy	size of grid spacing in y-direction (ft)
θ	dimensionless parameter between 0 and 1 used for generalized approach to applying boundary conditions
ρ	density (lb/ft^3)

Superscripts:

Δ	denotes temperature at a point outside the grid
n	previous time level
$n+1$	current time level
1	average with respect to the i-1,j node
2	average with respect to the i+1,j node
3	average with respect to the i,j+1 node
4	average with respect to the i,j-1 node

Subscripts:

1	denotes lower boundary for h and T; with respect to the i-1,j node for θ
2	denotes upper boundary for h and T; with respect to the i+1,j node for θ
3	denotes left boundary for h and T; with respect to the i,j+1 node for θ
4	denotes right boundary for h and T; with respect to the i,j-1 node for θ
i	grid point in the x-direction
$i1$	grid point in the x-direction which is evaluated with respect to the left layer
$i2$	grid point in the x-direction which is evaluated with respect to the right layer
j	grid point in the y- direction

j1	grid point in the y-direction which is evaluated with respect to the bottom layer
j2	grid point in the y-direction which is evaluated with respect to the top layer
l	liquid
lm	liquid at the melting point
m	melting point
s	solid
sm	solid at the melting point

ABSTRACT

Transient, numerical simulations of the deicing of composite aircraft components by electrothermal heating have been performed in a two-dimensional rectangular geometry. Seven numerical schemes and four solution methods were used to find the most efficient numerical procedure for this problem. The phase change in the ice was simulated using the Enthalpy method along with the Method of Assumed States. Numerical solutions illustrating deicer performance for various conditions are presented. Comparisons are made with previous numerical models and with experimental data. The simulation can also be used to solve a variety of other heat conduction problems involving composite bodies.

I. INTRODUCTION

The formation of ice on aircraft components during flight can severely affect aircraft performance. The ice presence increases drag and decreases lift which not only increases fuel consumption drastically but also jeopardizes the ability of the aircraft to fly. Aircraft which do not have any means of removing ice or preventing it from forming cannot fly in atmospheric conditions which promote ice accretion. As a result, there is a need to develop effective systems which can either keep ice from forming on these surfaces (anti-icing) or can remove the ice once it has formed (de-icing).

There are two commonly used systems for anti-icing aircraft components: chemical and thermal. Chemical systems lower the freezing point of water so that it will not freeze, much like anti-freeze in an automobile. However, these systems are of limited use during flight. Installation costs for supplying enough of the chemical during flight are prohibitive as are costs for maintenance of the system. Consequently, its use is mainly for pre-flight deicing and for windshield anti-icing.

Thermal systems prevent ice from forming by maintaining the surface temperature of the aircraft component above the melting point (32^{o}F). However, the energy requirement for this is quite high and is impractical for most aircraft. Jet airplanes use this method because of the availability of hot, compressed air from the engines.

There are three principal methods for de-icing aircraft components: thermal, mechanical and electroimpulse. Mechanical systems often employ a pneumatic boot which is laminated to the surface to be de-iced. The boot is a flexible, rubber-like material which, when inflated, breaks the ice off the surface. Boots are relatively simple and efficient, but require frequent maintenance to ensure reliability.

Electromechanical impulse is a new method still in development. A series of electromagnets are pulsed in cycles, flexing a metal abrasion shield and thereby mechanically cracking the ice. This method appears to be energy efficient, but it has had limited application to the deicing of aircraft components.

Electrothermal systems use electrical heater elements which are laminated to the surface to be deiced. These heater elements consist of metal ribbons which emit heat when an electrical currect is passed through them. This ribbon is surrounded by insulation and is covered with a top metal layer for protection. These heaters melt a small layer of ice at the surface which destroys the ice adhesion so that the aerodynamic forces can remove the ice from the surface. Energy requirements for de-icing are much less than for anti-icing, making this method practical for use on aircraft components.

Both electrothermal and pneumatic systems are used to deice aircraft components. However, pneumatic boots are not used on smaller aircraft because of the increase in drag caused when the boot inflates. This is especially true on helicopter blades. At this time, the electrothermal deicer system is considered to be the most effective means to deice helicopter blades.

The objective of this study is to develop an efficient numerical computer code which accurately predicts two-dimensional thermal transients in an electrothermal deicer pad. Various numerical methods will be studied to determine the most efficient method for this purpose. A comparison will be made to previous numerical codes and to existing experimental data. Finally, an effort will be made to determine how the parameters of the problem affect the solution and to determine under what circumstances a two-dimensional model is recommended over a one-dimensional model.

II. LITERATURE REVIEW

Recently, much work has been done in the area of electrothermal deicer pad design. This is especially true in the development of numerical models to simulate deicer pad performance. The first work in this area was performed by Stallabrass (1), who developed one and two-dimensional computer models. Baliga (2), Marano (3), Gent (4), and Roelke (5) all developed one-dimensional models using separate numerical techniques. Chao (6) developed a two-dimensional model which was later revised by Leffel (7). The present study considers a more complex two-dimensional problem than either Chao or Leffel considered and uses alternative solution methods. All of the above investigations were designed to model the phase change of the ice layer and to predict transient temperature distributions in composite layers.

Phase change, as first proposed by Stefan (1889), has been analyzed in numerous works over the years. Ockenden and Hodgkins (8) present an extensive review of most of the analytical and numerical techniques used in phase change analysis. The present study will focus on those methods which have previously been used for modeling phase change in deicer design.

Stallabrass (1) modeled the phase change by simply holding a node at the melting point until sufficient energy had accumulated to completely melt the node. Baliga (2) used a formulation proposed by Bonacina et al.(9), which associates the latent heat effect with a finite temperature interval about the phase change isotherm. The other models discussed previously all used a technique described by Voller and Cross (10,11) and is known as the enthalpy method. The enthalpy method is also termed a weak solution method because it is based on an integral formulation. In this procedure, temperature is calculated from the enthalpy and is therefore not calculated directly.

The enthalpy method uses the conservation of energy equation formulated in terms of two dependant variables, temperature and enthalpy. Predicting the location of the solid-liquid interface is not required because this is determined by nodal enthalpy alone. The temperature at any point is then calculated using the known enthalpy-temperature relationship. The equivalence of this method and the classical formulation of the ablation problem was proven by Atthey (12).

The previous investigations which used the enthalpy method employed a non-linear relationship between enthalpy and temperature during melting by keeping the melt temperature constant at 32 ^{o}F. This non-linearity requires that an iterative solution procedure be used in order to find the appropriate temperature at each point. In the present study, the enthapy method will be altered by the use of a small temperature interval at the melting point which has a high heat capacity similar to that used by Baliga (2). This adjustment is made to take advantage of a recent technique by Schneider and Raw (13) to predict temperatures during phase change in a more expedient fashion. Their technique assumes the phase at each point in the ice so that the enthalpy can be eliminated in favor of temperature prior to calculation. This in turn permits the use of non-iterative solution methods which allow the new temperature at each point to be determined more efficiently than by the use of iterative methods. The assumed phase of each node is then updated as the solution proceeds. This technique, which will be termed the Method of Assumed States, has been used by Roelke (5) in a one-dimensional deicer pad analysis.

Transient temperature distributions for more than two layers are extremely difficult to obtain except by numerical methods. Carslaw and Jaeger (14) used Laplace Transform techniques to solve two layer problems analytically, but for problems with more layers the inverse transform is extremely difficult to obtain. All of the models discussed previously use numerical methods to obtain their solutions.

Finite differencing is an approximate technique for solving boundary value problems. Carnahan, Luther and Wilkes (15) discuss various differencing methods that are commonly used and present both advantages and disadvantages. The finite difference method discretizes the continuous time and space domains into a grid of nodes. A system of algebraic equations based on this grid replaces the governing differential equations and boundary conditions. This system of equations can be solved to determine the value of the dependent variable at a particular node at any point in time and space. Accuracy of these methods depends on the particular method used and on the grid spacing. For example, the explicit method is first order accurate in time and second order accurate in space, whereas the Crank-Nicolson scheme is second order accurate in both time and space.

Many finite difference schemes can be found in the literature for the type of problem under study here. However, few of these have been applied to deicer modeling. Stallabrass (1) and Gent (4) both used an explicit method with forward-time, central-space differencing. Roelke (5) used a purely implicit method with forward-time, central-space differencing. Baliga (2), Marano (3), Chao (6), and Leffel(7) used the Crank-Nicolson scheme which is an implicit scheme that has central-time and central-space differencing. Implicit methods have been used to eliminate stability requirements of the explicit methods.

After finite differencing the equations, a variety of numerical techniques can be used to solve the corresponding matrix system of equations. The technique used may be determined by the finite differencing method, however. The simple explicit scheme needs no solution technique as all new temperatures are calculated from previously known values. For the one-dimensional heat conduction equation,this method requires that the time step (Δt) be less than $(\Delta x)^2/2\alpha$ to ensure stability, where α is the thermal diffusivity and Δt and Δx are the time and space increments, respectively.

The Crank-Nicolson scheme is implicit and therefore is often solved using an iterative solution technique. Baliga (2) used Gaussian elimination to solve for temperatures, which is very time consuming for a two-dimensional problem. Marano (3), Chao (6) and Leffel (7) used Gauss-Seidel iteration instead. This was used because of the non-linearity in the phase change procedure employed. Von Rosenburg (16) indicated that the Crank-Nicolson scheme is preferred over other methods for the type of differential equation that governs the problem under consideration here.

The present study will apply nine numerical techniques, some of which were developed after von Rosenburg's work, in order to determine which is the most efficient for modeling deicer pad designs. These methods were selected from those found in Anderson, Tannehill and Pletcher (17) to be the most promising for this problem. Three of these, the Simple Explicit, the Crank-Nicolson, and the Simple Implicit methods, have been used by previous investigators. The other six are the Hopscotch, ADE (Alternating Direction Explicit), ADI (Alternating Direction Implicit), time-splitting (the

Method of Fractional Steps), SIP (Strongly Implicit Procedure), and MSIP (Modified Strongly Implicit Procedure) methods.

The Simple Implicit method is quite similar to the Simple Explicit in that the differencing used is forward-time and central-space, but all of the temperatures are evaluated at the new time step instead of the old time step. It is unconditionally stable, as is Crank-Nicolson, and uses the same solution procedures as Crank-Nicolson. These three (Simple Explicit, Crank-Nicolson and Simple Implicit) are often grouped into a Combined method which will be described later. The Hopscotch and ADE methods are explicit, thus requiring no solution procedure for the resulting algebraic equations, but unlike the Simple Explicit method have no stability requirement. The ADE scheme used here was originally formulated by Barakat and Clark (18).

The other methods use the fact that the coefficient matrix for the temperatures, which arises from a particular finite difference scheme, is sparse in order to directly solve for temperatures, despite the fact that the differencing method is implicit. ADI, the oldest of these methods, was developed in 1955 by Peaceman and Rachford (19). It divides the time step into two parts and alternatively differences one spatial derivative implicitly and the other explicitly. Over a complete time step, the method is completely stable, as are all of the implicit schemes used here. A similar scheme to ADI was developed by Yanenko (20), which was called the Method of Fractional Steps. Anderson et al. (17) refer to this as a time-splitting method because it too divides the time step into two parts. For the two-dimensional diffusion equation, Yanenko proved that the scheme was equivalent to ADI. It is unknown whether this is still true when the phase change formulation is introduced. The SIP method was developed by Stone (21) and later improved upon by Schneider and Zedan (22) in their MSIP routine. Both of these methods can use any unconditionally stable implicit finite difference formulation and solve the matrix of temperatures which arise from this differencing procedure. In this study, Crank-Nicolson differencing was used because it has higher accuracy in the time domain than the Simple Implicit method. Although MSIP was shown by Schneider and Zedan to be faster than SIP, both were used here to confirm (or disprove) that finding.

III. EXTENSION OF THE TWO-DIMENSIONAL DEICER PROGRAM

This section will present the modifications which have been added to the existing two-dimensional deicer computer program. These modifications were added in order to simulate more complex deicing problems and to decrease the execution time of the simulation. These modifications include: variable properties in the heater layer; variable ice thickness with respect to both space and time coordinates; a variable number of heaters with variable firing sequences; a nonuniform outer heat transfer coefficient; noninsulated side boundary conditions; variable grid; and a significant decrease in computation time. The decrease in computation time was brought about mostly from the application of more efficient numerical techniques. These techniques have been briefly discussed in the previous section. All of the numerical techniques presented have incorporated the above modifications, but this discussion will focus on these improvements as they apply to the original Crank-Nicolson formulation.

First, improvements to the phase change layer will be discussed. Variable shapes of ice were obtained using a subroutine written by Leffel (7). His approach to variable ice shapes is detailed in his thesis and will not be repeated here. Variable ice growth rates were easily obtained by calling this subroutine at each time step and changing the amount of ice according to a rate specified in the data. Some modifications to the main program were necessary in order to accomodate parts of the exposed surface which did not have ice, i.e., those that had zero ice thickness and therefore would not undergo phase change. A variable heat transfer coefficient at the top surface of the ice was very easily implemented by specifying a distribution in the input data.

The other revisions in the ice layer were not implemented into the program which contains the various numerical methods to be discussed in the next section, but were implemented in separate variations of Chao's code (6). These programs were developed to study different ways to evaluate the conductivities in the ice layer during melting. Chao's code evaluated the conductivities using simple averages in both spatial coordinates. Marano (3), in the one-dimensional code he developed, evaluated the conductivities using an weighted average which he felt would more accurately model the phase change. By this method, the conductivities are evaluated using an average with the solid node above it if the control volume containing the node was less than one-half melted. Similarly, an average with the liquid node below would be used if the control volume containing the node were more than one-half melted. In this manner, melting would occur faster and less plateauing of the temperature profile would be evidenced. A version of Chao's program was developed which would evaluate conductivities in this manner in the y-direction only. The original technique was used in the x-direction. This was performed to show the agreement of Chao's code with Marano's for a one-dimensional case. The agreement between these two codes was exact to two decimal places.

The second program was an attempt to improve on the accuracy of these methods for evaluating conductivities in the phase change. The need for this arose from noticing that the nodes above the melt line sometimes produced small gradients which were in the opposite direction of the heat flow. That is, if in the rest of the grid heat flowed from right to left, these nodes would show a small temperature gradient which ran from left to right. Instead of using average conductivities, the conductivities were actually differentiated along with temperature. In equation form, this means

$$\partial/\partial x\ (\ k\ \partial T/\partial x\ ) + \partial/\partial y\ (\ k\ \partial T/\partial y\ ) = k\ \partial^2 T/\partial x^2 + k\ \partial^2 T/\partial y^2 + (\partial k/\partial x)\ (\partial T/\partial x) + (\partial k/\partial y)\ (\partial T/\partial y)$$

This was then centrally differenced using the Crank-Nicolson scheme and replaced the previous method of using average conductivities. The form of these equations will not be presented here, but their form is very similar to the original formulation. Computer runs using this formulation showed very little improvement of this effect. Hence all future improvements were performed using the original phase change formulation.

The next revisions to be discussed concern the heater layer. Originally, the only difference between the heaters and the gap was that the heater emitted energy while the gap did not. Hence, Chao's code simulated the heater gap as having the same properties and the same grid spacing as the heater itself. This is incorrect, since the space between heaters is generally composed of insulation, which has much different physical properties than the heater material. The program was modified so that a different material could be present in the heater gap and the grid spacing in the gap could be coarser or finer than the spacing in the heater. Since the temperature gradients are larger near the heater, finer mesh may be needed. The specifics of how the added interface is implemented will be presented in the last portion of the section on the application of boundary conditions.

Other modifications which have been added are: noninsulated side boundary conditions; over-relaxation of the Gauss-Seidel iteration procedure; changing the enthalpy method to accomodate the Method of Assumed States; and the use of numerical methods other than Crank-Nicolson as well as solution methods other than Gauss-Seidel iteration. Noninsulated side boundaries include nonzero heat transfer coefficients and constant temperature boundaries. Both of these are discussed in more detail in the section on application of boundary conditions. Over-relaxation is covered in the section on numerical solution methods, along with the other methods investigated. The next section will detail each of the finite difference methods used in the program.

IV. FORMULATION OF THE NUMERICAL METHOD

A. GOVERNING EQUATIONS AND BOUNDARY CONDITIONS

The following assumptions were made in the development of a two dimensional, transient, mathematical model for heat conduction in a composite blade with an ice layer:

(1) The thermal physical properties of the material composing each layer of the blade may be different, but do not depend on temperature;

(2) The ambient temperature, blade air temperature and all heat transfer coefficients are constant with respect to time;

(3) There is perfect thermal contact between each layer; and

(4) The density change due to melting is negligible, i.e., the effect of the volume contraction of the ice as it melts is neglected.

With the above assumptions, the mathematical formulation for the problem of unsteady heat conduction in a chordwise two-dimensional composite blade with electrothermal heating can be represented as

$$\rho_j C_{pj} (\partial T_j / \partial t) = k_j (\partial^2 T_j / \partial x^2 + \partial^2 T_j / \partial y^2) + q_j \qquad (1)$$

where j stands for the layer in question and where

ρ_j = density of the j^{th} layer;

C_{pj} = specific heat capacity of the j^{th} layer;

T_j = temperature in the j^{th} layer;

k_j = thermal conductivity of the j^{th} layer;

q_j = rate of heat generation per unit volume in the j^{th} layer;

t = time variable; and

x,y = spatial coordinates.

For the ice layer, the governing equation using the Enthalpy method is:

$$\partial H_{ice}/\partial t = \partial/\partial x\,[\,k_{ice}\,(\partial T_{ice}/\partial x)\,] + \partial/\partial y\,[k_{ice}\,(\partial T_{ice}/\partial y)\,] \qquad (2)$$

where

H_{ice} = Enthalpy per unit volume within the ice layer;

T_{ice} = Temperature within the ice layer; and

k_{ice} = thermal conductivity within the ice layer.

The enthalpy within both phases (ice and water) can be found from Eq. (2). The temperature can then be determined from the following relation between H and T:

$$H = \begin{cases} \rho_S C_{pS} T & T<T_m \\ \rho_L C_{pL} (T-T_m) + \rho_L (C_{pS} T_m + L_f) & T>T_m \end{cases} \qquad (3)$$

where

ρ_S, C_{pS} = physical properties of solid (ice) phase;

ρ_L, C_{pL} = physical properties of the liquid phase;

T_m = melting temperature; and

L_f = latent heat of fusion per unit mass.

From Eq. (3),

$$T = \begin{cases} H/\rho_s C_{ps} & H < H_{sm} \\ T_m & H_{sm} < H < H_{lm} \\ (H-H_{lm})/\rho_L C_{pL} + T_m & H > H_{lm} \end{cases} \tag{4}$$

with

$$H_{sm} = \rho_s C_{ps} T_m \tag{5a}$$

$$H_{lm} = \rho_L (C_{ps} T_m + L_f) \tag{5b}$$

where H_{sm} and H_{lm} are the enthalpies of the ice and water at the melting point, respectively.

The boundary and initial conditions are as follows:

(i) At interior interfaces, the temperatures and heat fluxes are continuous, i.e.,

$$T_{j1}/_{\text{interface}} = T_{j2}/_{\text{interface}} \tag{6}$$

$$-k_{j1} \; (\partial T_{j1}/\partial y_{j1})/_{\text{interface}} = -k_{j2} \; (\partial T_{j2}/\partial y_{j2})/_{\text{interface}} \tag{7}$$

For interior interfaces in the x-direction,

$$T_{i1}/_{\text{interface}} = T_{i2}/_{\text{interface}} \tag{8}$$

$$-k_{i1} \; (\partial T_{i1}/\partial x_{i1})/_{\text{interface}} = -k_{i2} \; (\partial T_{i2}/\partial x_{i2})/_{\text{interface}} \tag{9}$$

(ii) At the interior and exterior surfaces of the composite body, Newton's law of cooling can be used to represent the convective heat exchange.

For the lower boundary,

$$k_{j1} \; (\partial T_j/\partial y_j)_1 = h_{b1} \, (T_{j1} - T_{b1}) \tag{10}$$

where 1 denotes the lower or inner ambient boundary, h_{b1} is the convective heat transfer coefficient at the boundary and T_{b1} is the lower ambient (blade air) temperature.

For the upper or outer boundary,

$$-k_{j2}\ (\partial T_j/\partial y_j)_2 \ = \ h_{b2}\,(T_{j2} - T_{b2}) \tag{11}$$

where 2 denotes the upper or outer ambient boundary.

At the left boundary,

$$k_{j3}\ (\partial T_j/\partial x_j)_3 \ = \ h_{b3}\,(T_{j3} - T_{b3}) \tag{12}$$

where 3 denotes the left ambient boundary, h_{b3} is the convective heat transfer coefficient at the boundary and T_{b3} is the left ambient temperature. The coefficient h_{b3} can be set equal to zero to denote a line of symmetry.

At the right boundary,

$$-k_{j4}\ (\partial T_j/\partial x_j)_4 \ = \ h_{b4}\,(T_{j4} - T_{b4}) \tag{13}$$

where 4 denotes the right ambient boundary, h_{b4} is the convective heat transfer coefficient at the boundary and T_{b4} is the right ambient temperature. The coefficient h_{b4} can be set equal to zero to denote a line of symmetry. In addition to the above, constant temperature boundary conditions can be applied at all boundaries.

Next, the above equations will be expressed in finite difference form and arranged for numerical solution. A total of seven finite difference formulations will be presented.

B. FINITE DIFFERENCING PROCEDURES

Seven numerical procedures will be used in this thesis for the purpose of determining which is the most efficient for solving the equations described previously. Initially, a Crank-Nicolson finite differencing procedure was used and the resulting matrix of temperatures was solved by Gauss-Seidel iteration. At the time, this approach was used because it was thought that the non-linearity associated with the phase change would prohibit the use of direct solution methods which have been used for the two-dimensional diffusion equation. Presently, a technique has been found which will allow direct solution procedures to be used to solve this problem. Hence, an objective of the current investigation will be to examine these direct solution methods and determine which is the most efficient. The seven schemes to be studied are:

Simple Explicit
Crank-Nicolson
Simple Implicit
Hopscotch
ADE (Alternating Direction Explicit)
ADI (Alternating Direction Implicit)
Time-Splitting (also known as the Method of Fractional Steps)

1. Simple Explicit

The Simple Explicit method centrally differences the spatial coordinates and evaluates the temperatures at the previous time step to obtain a purely explicit method which has the stability requirement

$$\alpha\,\Delta t\,/(\,(\Delta x)^2 + (\Delta y)^2\,) \quad < \quad 1/2 \tag{14}$$

For Eq. (1) this becomes

$$\begin{aligned}(\,T^{n+1}_{i,j} - T^{n}_{i,j})/\Delta t = \alpha_{i,j}(\,T^{n}_{i+1,j} - 2\,T^{n}_{i,j} + T^{n}_{i-1,j})/(\Delta x_i)^2 \, + \\ \alpha_{i,j}(\,T^{n}_{i,j+1} - 2\,T^{n}_{i,j} + T^{n}_{i,j-1})\,/(\Delta y_j)^2\end{aligned} \tag{15}$$

For Eq. (2) this becomes

$$\begin{aligned}(\,H^{n+1}_{i,j} - H^{n}_{i,j})/\Delta t = k_{i,j}(\,T^{n}_{i+1,j} - 2\,T^{n}_{i,j} + T^{n}_{i-1,j})/(\Delta x_i)^2 \, + \\ k_{i,j}(\,T^{n}_{i,j+1} - 2\,T^{n}_{i,j} + T^{n}_{i,j-1})\,/(\Delta y_j)^2\end{aligned} \tag{16}$$

where

$\alpha_{i,j}$ = thermal diffusivity at node i,j ;

$\Delta t, \Delta x, \Delta y$ = time step and grid spacings;

n = the time step at which the solution is known.

Since in the ice layer, the conductivity may be a function of position due to the melting of the ice, average conductivities are used which are defined as

$$(a)\ k^1_{i,j} = (k_{i+1,j} + k_{i,j})/2. \quad (b)\ k^2_{i,j} = (k_{i-1,j} + k_{i,j})/2.$$

$$(c)\ k^3_{i,j} = (k_{i,j+1} + k_{i,j})/2. \quad (d)\ k^4_{i,j} = (k_{i,j-1} + k_{i,j})/2. \tag{17}$$

The previous equation for the ice layer can now be written as

$$(H^{n+1}_{i,j}-H^n_{i,j})/\Delta t = [k^1_{i,j}(T^n_{i+1,j} - T^n_{i,j}) + k^2_{i,j}(T^n_{i-1,j}-T^n_{i,j})]/(\Delta x_i)^2 + [k^3_{i,j}(T^n_{i,j+1} - T^n_{i,j}) + k^4_{i,j}(T^n_{i,j-1}-T^n_{i,j})]/(\Delta y_j)^2 \tag{18}$$

The boundary conditions given previously can also be finite differenced using this method. At an interface between layers,

x-direction interface:

$$-k_{i1,j}(T^n_{i+1,j} - T^n_{i-1,j})/2\Delta x_{i1} = -k_{i2,j}(T^n_{i+1,j} - T^n_{i-1,j})/2\Delta x_{i2} \tag{19a}$$

y-direction interface:

$$-k_{i,j1}(T^n_{i,j+1} - T^n_{i,j-1})/2\Delta y_{j1} = -k_{i,j2}(T^n_{i,j+1} - T^n_{i,j-1})/2\Delta y_{j2} \tag{19b}$$

For boundaries which have convective heat transfer coefficients,

$$k_{i,j}(T^n_{i+1,j} - T^n_{i-1,j})/2\Delta x_i = h_{b3}(T^n_{i,j} - T_{b3}) \tag{20a}$$

$$-k_{i,j}(T^n_{i+1,j} - T^n_{i-1,j})/2\Delta x_i = h_{b4}(T^n_{i,j} - T_{b4}) \tag{20b}$$

$$k_{i,j}(T^n_{i,j+1} - T^n_{i,j-1})/2\Delta y_j = h_{b1}(T^n_{i,j} - T_{b1}) \tag{20c}$$

$$-k_{i,j}(T^n_{i,j+1} - T^n_{i,j-1})/2\Delta y_j = h_{b2}(T^n_{i,j} - T_{b2}) \tag{20d}$$

These equations are then substituted into the governing equation for temperature at the appropriate boundary in order to eliminate temperatures which are outside of the matrix. The specifics of how this is done will be covered in a later section. The resulting matrix of temperatures can then be solved directly with no iteration. However, stability requirements can make the necessary time step so small as to make computation times prohibitively long. A standard deicer, for example, has such a small grid that the stability requirement for this method stipulates that the time step must be less than 10^{-4} seconds.

2. Crank-Nicolson

The Crank-Nicolson method is an implicit method which has been proven to be completely stable for the two-dimensional heat conduction equation. The spatial derivatives are again centrally differenced, but now they are evaluated at the n+1/2 time step. For Eq. (1) this becomes

$$\begin{aligned}(T^{n+1}_{i,j}-T^{n}_{i,j})/\Delta t &= \alpha_{i,j}(T^{n+1/2}_{i+1,j}-2\,T^{n+1/2}_{i,j}+T^{n+1/2}_{i-1,j})/(\Delta x_i)^2\\ &+ \alpha_{i,j}(T^{n+1/2}_{i,j+1}-2\,T^{n+1/2}_{i,j}+T^{n+1/2}_{i,j-1})/(\Delta y_j)^2\end{aligned}\tag{21}$$

The temperatures at the 1/2 time step can be evaluated using

$$T^{n+1/2} = (T^{n+1}+T^{n})/2\tag{22}$$

Equation (21) can thus be rewritten as

$$\begin{aligned}(T^{n+1}_{i,j}-T^{n}_{i,j})/\Delta t &= \alpha_{i,j}(T^{n+1}_{i+1,j}-2\,T^{n+1}_{i,j}+T^{n+1}_{i-1,j})/2(\Delta x_i)^2\\ &+ \alpha_{i,j}(T^{n+1}_{i,j+1}-2\,T^{n+1}_{i,j}+T^{n+1}_{i,j-1})/2(\Delta y_j)^2\\ &+ \alpha_{i,j}(T^{n}_{i+1,j}-2\,T^{n}_{i,j}+T^{n}_{i-1,j})/2(\Delta x_i)^2\\ &+ \alpha_{i,j}(T^{n}_{i,j+1}-2\,T^{n}_{i,j}+T^{n}_{i,j-1})/2(\Delta y_j)^2\end{aligned}\tag{23}$$

For Eq. (2), using the same average conductivities as presented for the Simple Explicit scheme, the finite difference form is

$$\begin{aligned}(H^{n+1}_{i,j}-H^{n}_{i,j})/\Delta t &= [\,k^1_{i,j}(T^{n+1}_{i+1,j}-T^{n+1}_{i,j})+k^2_{i,j}(T^{n+1}_{i-1,j}-T^{n+1}_{i,j})]/2(\Delta x_i)^2\\ &+ [\,k^3_{i,j}(T^{n+1}_{i,j+1}-T^{n+1}_{i,j})+k^4_{i,j}(T^{n+1}_{i,j-1}-T^{n+1}_{i,j})]/2(\Delta y_j)^2\\ &+ [\,k^1_{i,j}(T^{n}_{i+1,j}-T^{n}_{i,j})+k^2_{i,j}(T^{n}_{i-1,j}-T^{n}_{i,j})]/2(\Delta x_i)^2\\ &+ [\,k^3_{i,j}(T^{n}_{i,j+1}-T^{n}_{i,j})+k^4_{i,j}(T^{n}_{i,j-1}-T^{n}_{i,j})]/2(\Delta y_j)^2\end{aligned}\tag{24}$$

The boundary conditions given previously can also be finite differenced using the Crank-Nicolson method. At an interface between layers,

x-direction interface:

$$\begin{aligned}&-k_{i1,j}\,(T^{n+1}_{i+1,j}-T^{n+1}_{i-1,j})/4\Delta x_{i1}\ -k_{i1,j}\,(T^{n}_{i+1,j}-T^{n}_{i-1,j})/4\Delta x_{i1}\\&=-k_{i2,j}(T^{n+1}_{i+1,j}-T^{n+1}_{i-1,j})/4\Delta x_{i2}-k_{i2,j}(T^{n}_{i+1,j}-T^{n}_{i-1,j})/4\Delta x_{i2}\end{aligned}\tag{25a}$$

y-direction interface:

$$\begin{aligned}&-k_{i,j1}\,(T^{n+1}_{i,j+1}-T^{n+1}_{i,j-1})/4\Delta y_{j1}\ -k_{i,j1}(T^{n}_{i,j+1}-T^{n}_{i,j-1})/4\Delta y_{j1}\\&=-k_{i,j2}(T^{n+1}_{i,j+1}-T^{n+1}_{i,j-1})/4\Delta y_{j2}-k_{i,j2}(T^{n}_{i,j+1}-T^{n}_{i,j-1})/4\Delta y_{j2}\end{aligned}\tag{25b}$$

For boundaries which experience a convective heat transfer,

$$k_{i,j}(T^{n+1}_{i+1,j}-T^{n+1}_{i-1,j}+T^{n}_{i+1,j}-T^{n}_{i-1,j})/4\Delta x_i=h_{b3}((T^{n+1}_{i,j}+T^{n}_{i,j})/2-T_{b3})\tag{26a}$$

$$-k_{i,j}(T^{n+1}_{i+1,j}-T^{n+1}_{i-1,j}+T^{n}_{i+1,j}-T^{n}_{i-1,j})/4\Delta x_i=h_{b4}((T^{n+1}_{i,j}+T^{n}_{i,j})/2-T_{b4})\tag{26b}$$

$$k_{i,j}\,(T^{n+1}_{i,j+1}-T^{n+1}_{i,j-1}+T^{n}_{i,j+1}-T^{n}_{i,j-1})/4\Delta y_j=h_{b1}((T^{n+1}_{i,j}+T^{n}_{i,j})/2-T_{b1})\tag{26c}$$

$$-k_{i,j}\,(T^{n+1}_{i,j+1}-T^{n+1}_{i,j-1}+T^{n}_{i,j+1}-T^{n}_{i,j-1})/4\Delta y_j=h_{b2}((T^{n+1}_{i,j}+T^{n}_{i,j})/2-T_{b2})\tag{26d}$$

These equations are then substituted into the governing equation for temperature at the appropriate boundary in order to eliminate temperatures which are outside the matrix. The specifics of how this is accomplished will be covered in a later section.

The Crank-Nicolson algorithm is second order accurate in both time and spatial coordinates whereas the Simple Explicit scheme is only first order accurate in time and second order in space.

Since there is more than one unknown term in each of these equations, the temperature at the new time step can not be directly calculated. In this thesis, three methods will be discussed for solving the matrix of temperatures which arise from this differencing procedure. These three methods are Gauss-Seidel iteration, SIP (Strongly Implicit Procedure), and MSIP (Modified Strongly Implicit Procedure). They will be discussed in more detail in a subsequent chapter on numerical solution methods.

3. Simple Implicit

The Simple Implicit method also requires numerical solution by one of the three methods (Gauss-Seidel iteration, SIP, MSIP) mentioned above. It is unconditionally stable for all time steps, as was Crank-Nicolson, but it is only first order accurate in time and second order in space. This method is very similar to the Simple Explicit and Crank-Nicolson methods except that spatial derivatives are evaluated at the n+1 time step. These three methods are often grouped together into what is called a Combined Method since their forms are similar. The Simple Implicit finite difference form of Eq. (1) is

$$\begin{aligned}(T^{n+1}_{i,j}-T^{n}_{i,j})/\Delta t=\alpha_{i,j}(T^{n+1}_{i+1,j}-2T^{n+1}_{i,j}+T^{n+1}_{i-1,j})/(\Delta x_i)^2 &+ \\ \alpha_{i,j}(T^{n+1}_{i,j+1}-2T^{n+1}_{i,j}+T^{n+1}_{i,j-1})/(\Delta y_j)^2 & \end{aligned} \tag{27}$$

Similarly, for Eq. (2) using the previously defined average conductivities,

$$\begin{aligned}(H^{n+1}_{i,j}-H^{n}_{i,j})/\Delta t=[k^1_{i,j}(T^{n+1}_{i+1,j}-T^{n+1}_{i,j})+k^2_{i,j}(T^{n+1}_{i-1,j}-T^{n+1}_{i,j})]\,/(\Delta x_i)^2 \\ +[\,k^3_{i,j}\,(\,T^{n+1}_{i,j+1}-\,T^{n+1}_{i,j})+k^4_{i,j}\,(T^{n+1}_{i,j-1}-T^{n+1}_{i,j})]/(\Delta y_j)^2\end{aligned} \tag{28}$$

The boundary conditions given previously can also be finite differenced using this method. At an interface between layers,

x-direction interface:

$$-k_{i1,j}(T^{n+1}_{i+1,j}-T^{n+1}_{i-1,j})/2\Delta x_{i1} = -k_{i2,j}(T^{n+1}_{i+1,j} - T^{n+1}_{i-1,j})/2\Delta x_{i2} \tag{29a}$$

y-direction interface:

$$-k_{i,j1}(T^{n+1}_{i,j+1}-T^{n+1}_{i,j-1})/2\Delta y_{j1}=-k_{i,j2}(T^{n+1}_{i,j+1}-T^{n+1}_{i,j-1})/2\Delta y_{j2} \tag{29b}$$

For boundaries which have convective heat transfer coefficients,

$$k_{i,j}\,(T^{n+1}_{i+1,j} - T^{n+1}_{i-1,j})/2\Delta x_i = h_{b3}\,(T^{n+1}_{i,j} - T_{b3}) \tag{30a}$$

$$-\,k_{i,j}\,(T^{n+1}_{i+1,j} - T^{n+1}_{i-1,j})/2\Delta x_i = h_{b4}\,(T^{n+1}_{i,j} - T_{b4}) \tag{30b}$$

$$k_{i,j}\,(T^{n+1}_{i,j+1} - T^{n+1}_{i,j-1})/2\Delta y_j = h_{b1}\,(T^{n+1}_{i,j} - T_{b1}) \tag{30c}$$

$$-\,k_{i,j}\,(T^{n+1}_{i,j+1} - T^{n+1}_{i,j-1})/2\Delta y_j = h_{b2}(T^{n+1}_{i,j} - T_{b2}) \tag{30d}$$

These equations are then substituted into the governing equation for temperature at the appropriate boundary in order to eliminate temperatures which are outside of the coefficient matrix. The specifics of how this is done will be covered in a later section.

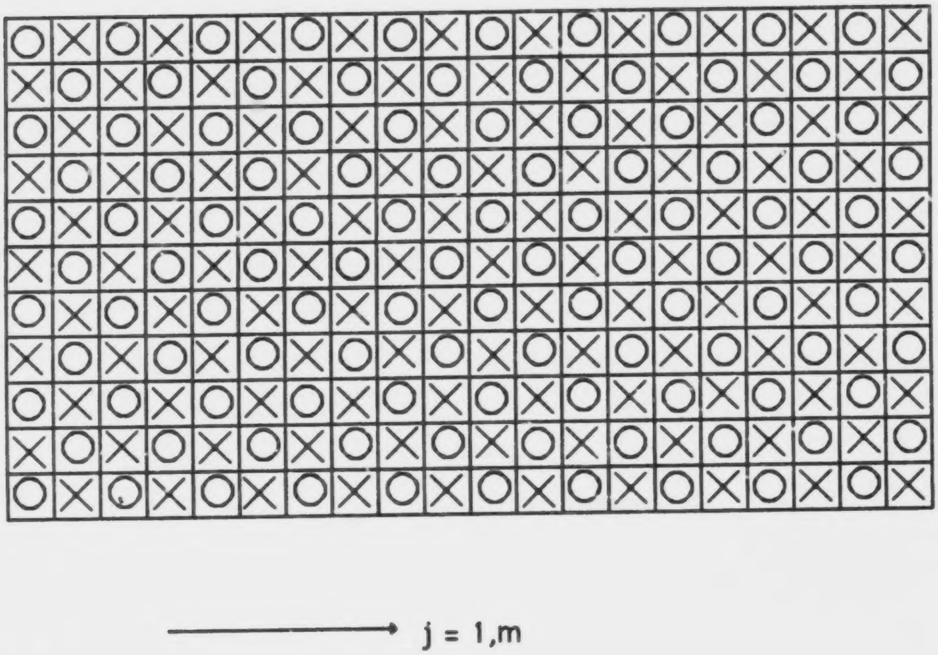
4. Hopscotch

The Hopscotch method is an explicit method which is unconditionally stable for the two-dimensional heat conduction equation. As such, it does not have a stability requirement as does the Simple Explicit method. The Hopscotch method is comprised of two steps. The first step evaluates the spatial derivatives explicitly but calculates temperatures only for those nodes in which the sum of i+j+n is even. The second step evaluates spatial derivatives implicitly, but calculates temperatures only for those nodes in which the sum of i+j+n is odd. This results in an explicit scheme since some of the temperatures in the second step were evaluated at the previous step.

Figure (1) shows the sequence of calculations described above. As can be seen, every node which is evaluated 'implicitly' is found using only those immediately next to it, all of which were found in the previous step. In this manner, all of the temperatures in the grid can be found for a particular time step. The finite difference equations used in the first step are the same as those for the Simple Explicit method, and those used in the second step are the same as for the Simple Implicit method. Hence, there is no unique finite difference formulation used in this method, but an interchange of the explicit and implicit methods from one node to the next.

Figure 1

Grid for Hopscotch Method



○	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×
×	○	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×	○
○	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×
×	○	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×	○
○	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×
×	○	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×	○
○	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×
×	○	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×	○
○	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×
×	○	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×	○
○	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×

○ denotes node which is evaluated explicitly at time n odd and implicitly at time n even.

× denotes node which is evaluated implicitly at time n odd and explicitly at time n even.

5. Alternating Direction Explicit (ADE)

The Alternating Direction Explicit method is an explicit method which, like the Hopscotch method, is unconditionally stable for all time steps. It is also comprised of two steps but, unlike the Hopscotch method, in this method every node is evaluated at each step and then the values obtained from these two steps are averaged. In each step, the spatial derivatives are centrally differenced, but are evaluated at different time steps. In the first step, the temperatures 'ahead' of node i,j are evaluated explicitly while those 'behind' node i,j are evaluated implicitly. Temperatures which are 'behind' node i,j have already been calculated in that pass, and hence are known values. For example, if the indices range from i=1 to N and j=1 to M, the temperatures at nodes i-1,j and i,j-1 have been previously calculated. The difference form of Eq. (1) for this case is

$$(T^{n+1}_{i,j}-T^{n}_{i,j})/\Delta t = \alpha_{i,j}[(T^{n}_{i+1,j}-T^{n}_{i,j})+(T^{n+1}_{i-1,j}-T^{n+1}_{i,j})]/(\Delta x_i)^2 + \alpha_{i,j}[(T^{n}_{i,j+1}-T^{n}_{i,j})+(T^{n+1}_{i,j-1}-T^{n+1}_{i,j})]/(\Delta y_j)^2 \tag{31}$$

The i index ranges from 1 to N and the j index ranges from 1 to M. As can be seen in the above equation, the only values evaluated at the new time step are at nodes i,j; i-1,j; and i,j-1. Since the values at i-1,j and i,j-1 were calculated previous to node i,j , the only unknown in this equation is at node i,j. Hence, the temperature can be calculated directly without iteration. The second step is the same, except the direction of calculation is reversed. For the second step,

$$(T^{n+1}_{i,j}-T^{n}_{i,j})/\Delta t = \alpha_{i,j}[(T^{n+1}_{i+1,j}-T^{n+1}_{i,j})+(T^{n}_{i-1,j}-T^{n}_{i,j})]/(\Delta x_i)^2 + \alpha_{i,j}[(T^{n+1}_{i,j+1}-T^{n+1}_{i,j})+(T^{n}_{i,j-1}-T^{n}_{i,j})]/(\Delta y_j)^2 \tag{32}$$

Here the i index ranges from N to 1 and the j index ranges from M to 1. Similarly, for the ice layer,

Step 1

$$(H^{n+1}_{i,j}-H^{n}_{i,j})/\Delta t=[k^1_{i,j}(T^{n}_{i+1,j}-T^{n}_{i,j})+k^2_{i,j}(T^{n+1}_{i-1,j}-T^{n+1}_{i,j})]/(\Delta x_i)^2 +[k^3_{i,j}(T^{n}_{i,j+1}-T^{n}_{i,j})+k^4_{i,j}(T^{n+1}_{i,j-1}-T^{n+1}_{i,j})]/(\Delta y_j)^2 \tag{33a}$$

Step 2

$$(H^{n+1}_{i,j}-H^{n}_{i,j})/\Delta t=[k^1_{i,j}(T^{n+1}_{i+1,j}-T^{n+1}_{i,j})+k^2_{i,j}(T^{n}_{i-1,j}-T^{n}_{i,j})]/(\Delta x_i)^2 +[k^3_{i,j}(T^{n+1}_{i,j+1}-T^{n+1}_{i,j})+k^4_{i,j}(T^{n}_{i,j-1}-T^{n}_{i,j})]/(\Delta y_j)^2 \tag{33b}$$

As stated earlier, the results of these two steps are averaged together to find the solution for the next time step. Each of these two steps individually is inconsistent since not all temperatures are evaluated at the same time step; however, together they represent a consistent scheme which is unconditionally stable.

The boundary conditions can also be finite differenced using this method. At an interface between layers,

x-direction interface: Step 1

$$-k_{i1,j}\,(T^{n}_{i+1,j} - T^{n+1}_{i-1,j})/2\Delta x_{i1} = -k_{i2,j}\,(T^{n}_{i+1,j} - T^{n+1}_{i-1,j})/2\Delta x_{i2} \qquad (34a)$$

Step 2

$$-k_{i1,j}\,(T^{n+1}_{i+1,j} - T^{n}_{i-1,j})/2\Delta x_{i1} = -k_{i2,j}\,(T^{n+1}_{i+1,j} - T^{n}_{i-1,j})/2\Delta x_{i2} \qquad (34b)$$

y-direction interface: Step 1

$$-k_{i,j1}\,(T^{n}_{i,j+1} - T^{n+1}_{i,j-1})/2\Delta y_{j1} = -k_{i,j2}(T^{n}_{i,j+1} - T^{n+1}_{i,j-1})/2\Delta y_{j2} \qquad (34c)$$

Step 2

$$-k_{i,j1}\,(T^{n+1}_{i,j+1} - T^{n}_{i,j-1})/2\Delta y_{j1} = -k_{i,j2}(T^{n+1}_{i,j+1} - T^{n}_{i,j-1})/2\Delta y_{j2} \qquad (34d)$$

For boundaries which have a convective heat exchange,

Step 1

$$k_{i,j}(T^{n}_{i+1,j}-T^{n+1}_{i-1,j})/2\Delta x_i = h_{b3}((T^{n+1}_{i,j}+T^{n}_{i,j})/2-T_{b3}) \qquad (35a)$$

$$-k_{i,j}(T^{n}_{i+1,j}-T^{n+1}_{i-1,j})/2\Delta x_i = h_{b4}((T^{n+1}_{i,j}+T^{n}_{i,j})/2-T_{b4}) \qquad (35b)$$

$$k_{i,j}(T^{n}_{i,j+1}-T^{n+1}_{i,j-1})/2\Delta y_j = h_{b1}((T^{n+1}_{i,j}+T^{n}_{i,j})/2-T_{b1}) \qquad (35c)$$

$$-k_{i,j}(T^{n}_{i,j+1}-T^{n+1}_{i,j-1})/2\Delta y_j = h_{b2}((T^{n+1}_{i,j}+T^{n}_{i,j})/2-T_{b2}) \qquad (35d)$$

Step 2

$$k_{i,j}(T^{n+1}{}_{i+1,j}-T^{n}{}_{i-1,j})/2\Delta x_i=h_{b3}((T^{n+1}{}_{i,j}+T^{n}{}_{i,j})/2-T_{b3}) \qquad (35e)$$

$$-k_{i,j}(T^{n+1}{}_{i+1,j}-T^{n}{}_{i-1,j})/2\Delta x_i=h_{b4}((T^{n+1}{}_{i,j}+T^{n}{}_{i,j})/2-T_{b4}) \qquad (35f)$$

$$k_{i,j}(T^{n+1}{}_{i,j+1}-T^{n}{}_{i,j-1})/2\Delta y_j=h_{b1}((T^{n+1}{}_{i,j}+T^{n}{}_{i,j})/2-T_{b1}) \qquad (35g)$$

$$-k_{i,j}(T^{n+1}{}_{i,j+1}-T^{n}{}_{i,j-1})/2\Delta y_j=h_{b2}((T^{n+1}{}_{i,j}+T^{n}{}_{i,j})/2-T_{b2}) \qquad (35h)$$

6. Alternating Direction Implicit (ADI)

The Alternating Direction Implicit method uses a formulation which splits the time step into two parts. In the first half time step, one of the spatial derivatives is evaluated explicitly and the other is evaluated implicitly. For the second half time step, the spatial derivative which was evaluated explicitly is now evaluated implicitly and the one which was evaluated implicitly is now evaluated explicitly. This process is then repeated for the duration of the simulation. Explicitly evaluating the x-derivative first results in the following finite difference equations:

First Half Time Step

$$(T^{n+1/2}{}_{i,j}-T^{n}{}_{i,j})/\Delta t = \alpha_{i,j}[(T^{n}{}_{i+1,j}-T^{n}{}_{i,j})+(T^{n}{}_{i-1,j}-T^{n}{}_{i,j})]/2(\Delta x_i)^2 + \alpha_{i,j}[(T^{n+1/2}{}_{i,j+1}-T^{n+1/2}{}_{i,j})+(T^{n+1/2}{}_{i,j-1}-T^{n+1/2}{}_{i,j})]/2(\Delta y_j)^2 \qquad (36a)$$

Second Half Time Step

$$(T^{n+1}{}_{i,j}-T^{n+1/2}{}_{i,j})/\Delta t=\alpha_{i,j}[(T^{n+1}{}_{i+1,j}-T^{n+1}{}_{i,j})+(T^{n+1}{}_{i-1,j}-T^{n+1}{}_{i,j})]/2(\Delta x_i)^2 +\alpha_{i,j}[(T^{n+1/2}{}_{i,j+1}-T^{n+1/2}{}_{i,j})+(T^{n+1/2}{}_{i,j-1}-T^{n+1/2}{}_{i,j})]/2(\Delta y_j)^2 \qquad (36b)$$

For the ice layer equation,

First Half Time Step

$$(H^{n+1/2}_{i,j}-H^{n}_{i,j})/\Delta t=[k^1_{i,j}(T^{n}_{i+1,j}-T^{n}_{i,j})+k^2_{i,j}(T^{n}_{i-1,j}-T^{n}_{i,j})]/2(\Delta x_i)^2$$

$$+[k^3_{i,j}(T^{n+1/2}_{i,j+1}-T^{n+1/2}_{i,j})+k^4_{i,j}(T^{n+1/2}_{i,j-1}-T^{n+1/2}_{i,j})]/2(\Delta y_j)^2 \tag{37a}$$

Second Half Time Step

$$(H^{n+1}_{i,j}-H^{n+1/2}_{i,j})/\Delta t=[k^1_{i,j}(T^{n+1}_{i+1,j}-T^{n+1}_{i,j})+k^2_{i,j}(T^{n+1}_{i-1,j}-T^{n+1}_{i,j})]/2(\Delta x_i)^2$$

$$+[k^3_{i,j}(T^{n+1/2}_{i,j+1}-T^{n+1/2}_{i,j})+k^4_{i,j}(T^{n+1/2}_{i,j-1}-T^{n+1/2}_{i,j})]/2(\Delta y_j)^2 \tag{37b}$$

Similarly, the boundary conditions can also be finite differenced using tnis method. At an interface between layers,

x-direction interface: Step 1

$$-k_{i1,j}(T^{n+1/2}_{i+1,j}-T^{n+1/2}_{i-1,j})/2\Delta x_{i1}=-k_{i2,j}(T^{n+1/2}_{i+1,j}-T^{n+1/2}_{i-1,j})/2\Delta x_{i2} \tag{38a}$$

Step 2

$$-k_{i1,j}(T^{n+1/2}_{i+1,j}-T^{n+1/2}_{i-1,j})/2\Delta x_{i1}=-k_{i2,j}(T^{n+1/2}_{i+1,j}-T^{n+1/2}_{i-1,j})/2\Delta x_{i2} \tag{38b}$$

y-direction interface: Step 1

$$-k_{i,j1}(T^{n}_{i,j+1}-T^{n}_{i,j-1})/2\Delta y_{j1}=-k_{i,j2}(T^{n}_{i,j+1}-T^{n}_{i,j-1})/2\Delta y_{j2} \tag{38c}$$

Step 2

$$-k_{i,j1}(T^{n+1}_{i,j+1}-T^{n+1}_{i,j-1})/2\Delta y_{j1}=-k_{i,j2}(T^{n+1}_{i,j+1}-T^{n+1}_{i,j-1})/2\Delta y_{j2} \tag{38d}$$

For boundaries which have convective heat transfer coefficients,

Step 1

$$k_{i,j}(T^{n+1/2}_{i+1,j}-T^{n+1/2}_{i-1,j})/2\Delta x_i=h_{b3}(T^{n+1/2}_{i,j}-T_{b3}) \quad (39a)$$

$$-k_{i,j}(T^{n+1/2}_{i+1,j}-T^{n+1/2}_{i-1,j})/2\Delta x_i=h_{b4}(T^{n+1/2}_{i,j}-T_{b4}) \quad (39b)$$

$$k_{i,j}(T^{n}_{i,j+1}-T^{n}_{i,j-1})/2\Delta y_j = h_{b1}(T^{n}_{i,j}-T_{b1}) \quad (39c)$$

$$-k_{i,j}(T^{n}_{i,j+1}-T^{n}_{i,j-1})/2\Delta y_j = h_{b2}(T^{n}_{i,j}-T_{b2}) \quad (39d)$$

Step 2

$$k_{i,j}(T^{n+1/2}_{i+1,j}-T^{n+1/2}_{i-1,j})/2\Delta x_i=h_{b3}(T^{n+1/2}_{i,j}-T_{b3}) \quad (39e)$$

$$-k_{i,j}(T^{n+1/2}_{i+1,j}-T^{n+1/2}_{i-1,j})/2\Delta x_i=h_{b4}(T^{n+1/2}_{i,j}-T_{b4}) \quad (39f)$$

$$k_{i,j}(T^{n+1}_{i,j+1}-T^{n+1}_{i,j-1})/2\Delta y_j = h_{b1}(T^{n+1}_{i,j}-T_{b1}) \quad (39g)$$

$$-k_{i,j}(T^{n+1}_{i,j+1}-T^{n+1}_{i,j-1})/2\Delta y_j = h_{b2}(T^{n+1}_{i,j}-T_{b2}) \quad (39h)$$

Except for the ice layer, the only unknown values are the temperatures at nodes i,j; i+1,j; and i-1,j for the first time step and i,j; i,j+1; i,j-1 for the second time step. In each case, a tridiagonal system of equations is produced which can easily be inverted to find a solution. For the ice layer, all that needs to be done is to eliminate enthalpy in favor of temperature so that the solution can be found in the same manner as for the rest of the grid. A technique for accomplishing this will be discussed in a later section. This method is unconditionally stable and completely consistent after two time steps have been completed.

7. Time-Splitting

The Time-Splitting or Method of Fractional Steps is an implicit method which is unconditionally stable and completely consistent after two time steps. Like ADI, the algorithm needs two time steps in order to be consistent. It consists of neglecting one of the spatial derivatives in the first time step and then neglecting the other in the second time step. The resulting one-dimensional equations are then differenced using either the Crank-Nicolson or Simple Implicit methods. Using Crank-Nicolson differencing, the time-split form of Eq. (1) is

First Half Time Step

$$
\begin{aligned}
(T^{n+1/2}_{i,j}-T^{n}_{i,j})/\Delta t &= \alpha_{i,j}[(T^{n}_{i+1,j}-T^{n}_{i,j})+(T^{n}_{i-1,j}-T^{n}_{i,j})]/2(\Delta x_i)^2 \\
&+\alpha_{i,j}[(T^{n+1/2}_{i+1,j}-T^{n+1/2}_{i,j})+(T^{n+1/2}_{i-1,j}-T^{n+1/2}_{i,j})]/2(\Delta x_i)^2
\end{aligned}
\tag{40a}
$$

Second Half Time Step

$$
\begin{aligned}
(T^{n+1}_{i,j}-T^{n+1/2}_{i,j})/\Delta t &= \alpha_{i,j}[(T^{n+1}_{i,j+1}-T^{n+1}_{i,j})+(T^{n+1}_{i,j-1}-T^{n+1}_{i,j})]/2(\Delta y_j)^2 \\
&+\alpha_{i,j}[(T^{n+1/2}_{i,j+1}-T^{n+1/2}_{i,j})+(T^{n+1/2}_{i,j-1}-T^{n+1/2}_{i,j})]/2(\Delta y_j)^2
\end{aligned}
\tag{40b}
$$

For the ice layer equation,

First Half Time Step

$$
\begin{aligned}
(H^{n+1}_{i,j}-H^{n+1/2}_{i,j})/\Delta t &= [k^1_{i,j}(T^{n+1}_{i,j+1}-T^{n+1}_{i,j})+k^2_{i,j}(T^{n+1}_{i,j-1}-T^{n+1}_{i,j})]/2(\Delta x_i)^2 \\
&+[k^1_{i,j}(T^{n+1/2}_{i+1,j}-T^{n+1/2}_{i,j})+k^2_{i,j}(T^{n+1/2}_{i-1,j}-T^{n+1/2}_{i,j})]/2(\Delta x_i)^2
\end{aligned}
\tag{41a}
$$

Second Half Time Step

$$
\begin{aligned}
(H^{n+1}_{i,j}-H^{n+1/2}_{i,j})/\Delta t &= [k^3_{i,j}(T^{n+1}_{i,j+1}-T^{n+1}_{i,j})+k^4_{i,j}(T^{n+1}_{i,j-1}-T^{n+1}_{i,j})]/2(\Delta y_j)^2 \\
&+[k^3_{i,j}(T^{n+1/2}_{i,j+1}-T^{n+1/2}_{i,j})+k^4_{i,j}(T^{n+1/2}_{i,j-1}-T^{n+1/2}_{i,j})]/2(\Delta y_j)^2
\end{aligned}
\tag{41b}
$$

This formulation also results in a tridiagonal coefficient matrix which can be easily inverted. The Crank-Nicolson difference form of this method was presented here since this was the form used in all simulations. The finite difference form of the boundary conditions for this method is the same as for the Crank-Nicolson method. The next section will deal with how the finite difference form of the boundary conditions for each of these methods are applied to the finite difference form of the governing equation.

C. APPLICATION OF BOUNDARY CONDITIONS

When a boundary is reached in the grid, any of the finite difference forms of the heat conduction equation in the previous section will have temperatures which are not within the grid. These temperatures are eliminated from the equation using one of the boundary conditions which have previously been differenced. In this section, this elimination process will be presented in a general format. The general format used is an expansion of the Combined method mentioned earlier. The Combined method is a combination of the Simple Explicit, Crank-Nicolson, and Simple Implicit methods where theta (θ) is a parameter between 0 and 1 that is used to determine the method to be employed. For $\theta = 0$, the Explicit method is obtained; for $\theta = 1/2$, the Crank-Nicolson method is obtained; and for $\theta = 1$ the Implicit method is obtained. For the two-dimensional heat conduction equation, this becomes

$$
\begin{aligned}
(T^{n+1}_{i,j}-T^{n}_{i,j})/\Delta t = {} & \alpha_{i,j}(T^{n+1}_{i+1,j}-2T^{n+1}_{i,j}+T^{n+1}_{i-1,j})\,\theta/(\Delta x_i)^2 \\
& +\alpha_{i,j}(T^{n+1}_{i,j+1}-2T^{n+1}_{i,j}+T^{n+1}_{i,j-1})\,\theta/(\Delta y_j)^2 \\
& +\alpha_{i,j}(T^{n}_{i+1,j}-2T^{n}_{i,j}+T^{n}_{i-1,j})(1-\theta)/(\Delta x_i)^2 \\
& +\alpha_{i,j}(T^{n}_{i,j+1}-2T^{n}_{i,j}+T^{n}_{i,j-1})(1-\theta)/(\Delta y_j)^2
\end{aligned}
\qquad (42)
$$

The extension of this to all of the methods presented previously can be accomplished by introducing four parameters (θ_1, θ_2, θ_3, and θ_4) which are all between 0 and 1. The equations which follow represent the governing equations for the composite blade, the ice layer, and boundary conditions. Table1 provides values of θ_1, θ_2, θ_3, and θ_4 for each of the methods presented previously.

Table 1

Theta Values for Each Numerical Method

Method	Theta 1	Theta 2	Theta 3	Theta 4
Simple Explicit	0.0	0.0	0.0	0.0
Crank-Nicolson	0.5	0.5	0.5	0.5
Simple Implicit	1.0	1.0	1.0	1.0
Hopscotch (i + j + n even)	0.0	0.0	0.0	0.0
Hopscotch (i + j + n odd)	1.0	1.0	1.0	1.0
ADE (first pass)	0.0	1.0	0.0	1.0
ADE (second pass)	1.0	0.0	1.0	0.0
ADI (first time step)	0.0	0.0	1.0	1.0
ADI (second time step)	1.0	1.0	0.0	0.0
Time-Splitting (first time step)	0.5	0.5	0.0 *	0.0 *
Time-Splitting (second time step)	0.0 *	0.0 *	0.5	0.5

* -- the quantity $(1-\theta)=0$ as well for this method.

For the composite blade,

$$
\begin{aligned}
(T^{n+1}_{i,j}-T^{n}_{i,j})/\Delta t=\alpha_{i,j}&[\theta_1(T^{n+1}_{i+1,j}-T^{n+1}_{i,j})+\theta_2(T^{n+1}_{i-1,j}-T^{n+1}_{i,j})]/(\Delta x_i)^2\\
&+\alpha_{i,j}[\theta_3(T^{n+1}_{i,j+1}-T^{n+1}_{i,j})+\theta_4(T^{n+1}_{i,j-1}-T^{n+1}_{i,j})]/(\Delta y_j)^2\\
&+\alpha_{i,j}[(1-\theta_1)(T^{n}_{i+1,j}-T^{n}_{i,j})+(1-\theta_2)(T^{n}_{i-1,j}-T^{n}_{i,j})]/(\Delta x_i)^2\\
&+\alpha_{i,j}[(1-\theta_3)(T^{n}_{i,j+1}-T^{n}_{i,j})+(1-\theta_4)(T^{n}_{i,j-1}-T^{n}_{i,j})]/(\Delta y_j)^2
\end{aligned}
\tag{43}
$$

For the ice layer,

$$\begin{aligned}
&(H^{n+1}_{i,j}-H^{n}_{i,j})/\Delta t=[\theta_1 k^1_{i,j}(T^{n+1}_{i+1,j}-T^{n+1}_{i,j})+\theta_2 k^2_{i,j}(T^{n+1}_{i-1,j}-T^{n+1}_{i,j})]/(\Delta x_i)^2\\
&+[\theta_3 k^3_{i,j}(T^{n+1}_{i,j+1}-T^{n+1}_{i,j})+\theta_4 k^4_{i,j}(T^{n+1}_{i,j-1}-T^{n+1}_{i,j})]/(\Delta y_j)^2\\
&+[(1-\theta_1)k^1_{i,j}(T^{n}_{i+1,j}-T^{n}_{i,j})+(1-\theta_2)k^2_{i,j}(T^{n}_{i-1,j}-T^{n}_{i,j})]/(\Delta x_i)^2\\
&+[(1-\theta_3)k^3_{i,j}(T^{n}_{i,j+1}-T^{n}_{i,j})+(1-\theta_4)k^4_{i,j}(T^{n}_{i,j-1}-T^{n}_{i,j})]/(\Delta y_j)^2
\end{aligned}\tag{44}$$

x-direction interface

$$\begin{aligned}
&-k_{i1,j}(\theta_1 T^{n+1}_{i+1,j}-\theta_2 T^{n+1}_{i-1,j})/4\Delta x_{i1}-k_{i1,j}((1-\theta_1)T^{n}_{i+1,j}-(1-\theta_2)T^{n}_{i-1,j})/4\Delta x_{i1}\\
&=-k_{i2,j}(\theta_1 T^{n+1}_{i+1,j}-\theta_2 T^{n+1}_{i-1,j})/4\Delta x_{i2}-k_{i2,j}((1-\theta_1)T^{n}_{i+1,j}-(1-\theta_2)T^{n}_{i-1,j})/4\Delta x_{i2}
\end{aligned}\tag{45a}$$

y-direction interface

$$\begin{aligned}
&-k_{i,j1}(\theta_3 T^{n+1}_{i,j+1}-\theta_4 T^{n+1}_{i,j-1})/4\Delta y_{j1}-k_{i,j1}((1-\theta_3)T^{n}_{i,j+1}-(1-\theta_4)T^{n}_{i,j-1})/4\Delta y_{j1}\\
&=-k_{i,j2}(\theta_3 T^{n+1}_{i,j+1}-\theta_4 T^{n+1}_{i,j-1})/4\Delta y_{j2}-k_{i,j2}((1-\theta_3)T^{n}_{i,j+1}-(1-\theta_4)T^{n}_{i,j-1})/4\Delta y_{j2}
\end{aligned}\tag{45b}$$

For boundaries which have convective heat transfer coefficients,

$$\begin{aligned}
&k_{i,j}(\theta_1 T^{n+1}_{i+1,j}-\theta_2 T^{n+1}_{i-1,j}+(1-\theta_1)T^{n}_{i+1,j}-(1-\theta_2)T^{n}_{i-1,j})/4\Delta x_i\\
&=h_{b3}(((\theta_1+\theta_2)T^{n+1}_{i,j}+(2-\theta_1-\theta_2)T^{n}_{i,j})-2T_{b3})/2
\end{aligned}\tag{46a}$$

$$\begin{aligned}
&-k_{i,j}(\theta_1 T^{n+1}_{i+1,j}-\theta_2 T^{n+1}_{i-1,j}+(1-\theta_1)T^{n}_{i+1,j}-(1-\theta_2)T^{n}_{i-1,j})/4\Delta x_i\\
&=h_{b4}(((\theta_1+\theta_2)T^{n+1}_{i,j}+(2-\theta_1-\theta_2)T^{n}_{i,j})-2T_{b4})/2
\end{aligned}\tag{46b}$$

$$\begin{aligned}
&k_{i,j}(\theta_3 T^{n+1}_{i,j+1}-\theta_4 T^{n+1}_{i,j-1}+(1-\theta_3)T^{n}_{i,j+1}-(1-\theta_4)T^{n}_{i,j-1})/4\Delta y_j\\
&=h_{b1}(((\theta_3+\theta_4)T^{n+1}_{i,j}+(2-\theta_3-\theta_4)T^{n}_{i,j})-2T_{b1})/2
\end{aligned}\tag{46c}$$

$$-k_{i,j}(\theta_3 T^{n+1}_{i,j+1}-\theta_4 T^{n+1}_{i,j-1}+(1-\theta_3)T^{n}_{i,j+1}-(1-\theta_4)T^{n}_{i,j-1})/4\Delta y_j$$

$$=h_{b2}(((\theta_3+\theta_4)T^{n+1}_{i,j}+(2-\theta_3-\theta_4)T^{n}_{i,j})-2T_{b2})/2 \qquad (46d)$$

The boundary conditions can now be applied in this general format to obtain generalized equations for all points in the grid. It should be noted that since the boundary conditions have been differenced so as to be compatible with the heat conduction equation, the statements made earlier about the methods of solution are still applicable.

First, boundary conditions will be applied to the ice layer. In this layer, there are no x-direction interfaces between layers, nor is the inner heat transfer coefficient applicable. Hence, the derivations given here will deal only with the remaining boundary condition equations. For heat transfer at the top surface of the ice, Eqs. (45) and (46c) are used to eliminate temperatures $T^{n+1}_{i,j+1}$ and $T^{n}_{i,j+1}$, which are not part of the computational grid. The superscript $^{\Delta}$ will denote this fictitious temperature. Solving Eq. (46c) for these temperatures yields

$$k^{3}_{i,j}(\theta_3 T^{n+1}_{i,j+1}+(1-\theta_3)T^{n}_{i,j+1})^{\Delta}/4\Delta y_j= k^{4}_{i,j}(\theta_4 T^{n+1}_{i,j-1}+(1-\theta_4)T^{n}_{i,j-1})/4\Delta y_j$$

$$-h_{b2}(((\theta_3+\theta_4)T^{n+1}_{i,j}+(2-\theta_3-\theta_4)T^{n}_{i,j})-2T_{b2})/2 \qquad (47)$$

Substitution of this equation into Eq. (45) to eliminate $T^{n+1}_{i,j+1}$ and $T^{n}_{i,j+1}$ yields

$$(H^{n+1}_{i,j}-H^{n}_{i,j})/\Delta t=[\theta_1 k^{1}_{i,j}(T^{n+1}_{i+1,j}-T^{n+1}_{i,j})+\theta_2 k^{2}_{i,j}(T^{n+1}_{i-1,j}-T^{n+1}_{i,j})]/(\Delta x_i)^2$$

$$+[2\theta_3 k^{3}_{i,j}(T^{n+1}_{i,j+1}-T^{n+1}_{i,j})+2(1-\theta_3)k^{3}_{i,j}(T^{n}_{i,j+1}-T^{n}_{i,j})]/(\Delta y_j)^2$$

$$+[(1-\theta_1)k^{1}_{i,j}(T^{n}_{i+1,j}-T^{n}_{i,j})+(1-\theta_2)k^{2}_{i,j}(T^{n}_{i-1,j}-T^{n}_{i,j})]/(\Delta x_i)^2 \qquad (48)$$

$$-h_{b2}(((\theta_3+\theta_4)T^{n+1}_{i,j}+(2-\theta_3-\theta_4)T^{n}_{i,j})-2T_{b2})/\Delta y_j$$

Rearranging this equation in order to have all $T^{n+1}_{i,j}$ terms on the left-hand side results in the following:

$$H^{n+1}_{i,j}+\Delta t\ T^{n+1}_{i,j}((\theta_1 k^1_{i,j}+\theta_2 k^2_{i,j})/\Delta x^2_i+(\theta_3 k^3_{i,j}+\theta_4 k^4_{i,j})/(\Delta y_j)^2\ +h_{b2}(\theta_3+\theta_4)/\Delta y_j)$$

$$=H^n_{i,j}+\Delta t\ [(\theta_1 k^1_{i,j}T^{n+1}_{i+1,j}+\theta_2 k^2_{i,j}T^{n+1}_{i-1,j})\ /(\Delta x_i)^2)$$

$$+(2\theta_3 k^3_{i,j}T^{n+1}_{i,j+1}+2(1-\theta_3)k^3_{i,j}T^n_{i,j+1})/(\Delta y_j)^2 \qquad (49)$$

$$+[(1-\theta_1)k^1_{i,j}(T^n_{i+1,j}-T^n_{i,j})+(1-\theta_2)k^2_{i,j}(T^n_{i-1,j}-T^n_{i,j})]/(\Delta x_i)^2$$

$$-T^n_{i,j}((1-\theta_3)k^3_{i,j}+(1-\theta_4)k^4_{i,j})/(\Delta y_j)^2-h_{b2}((2-\theta_3-\theta_4)T^n_{i,j}/2-2T_{b2})/\Delta y_j]$$

This equation applies to the case in which there is a convective heat exchange at the top surface of the ice layer. When heat transfer occurs on the left surface of the ice, Eq. (46a) is first solved to determine the fictitious temperatures $T^{n+1}_{i-1,j}$ and $T^n_{i-1,j}$. These are then substituted into Eq. (45) in the same manner as described above to obtain

$$H^{n+1}_{i,j}+\Delta t\ T^{n+1}_{i,j}((\theta_1 k^1_{i,j}+\theta_2 k^2_{i,j})/(\Delta x_i)^2+(\theta_3 k^3_{i,j}+\theta_4 k^4_{i,j})/(\Delta y_j)^2\ +h_{b3}(\theta_1+\theta_2)/\Delta x_i)$$

$$=H^n_{i,j}+\Delta t[(\theta_3 k^3_{i,j}T^{n+1}_{i,j+1}+\theta_4 k^4_{i,j}T^{n+1}_{i,j-1})/(\Delta y_j)^2)$$

$$+(2\theta_1 k^1_{i,j}T^{n+1}_{i+1,j}+2(1-\theta_1)k^1_{i,j}T^n_{i+1,j})/(\Delta x_i)^2 \qquad (50)$$

$$+[(1-\theta_3)k^3_{i,j}(T^n_{i,j+1}-T^n_{i,j})+(1-\theta_4)k^4_{i,j}(T^n_{i,j-1}-T^n_{i,j})]/(\Delta y_j)^2$$

$$-T^n_{i,j}((1-\theta_1)k^1_{i,j}+(1-\theta_2)k^2_{i,j})/(\Delta x_i)^2-h_{b3}((2-\theta_1-\theta_2)T^n_{i,j}-2T_{b3})/\Delta x_i]$$

Similarly, where there is heat transfer on the right surface,

$$H^{n+1}_{i,j}+\Delta t\ T^{n+1}_{i,j}((\theta_1 k^1_{i,j}+\theta_2 k^2_{i,j})/(\Delta x_i)^2+(\theta_3 k^3_{i,j}+\theta_4 k^4_{i,j})/(\Delta y_j)^2+h_{b4}(\theta_1+\theta_2)/\Delta x_i)$$

$$=H^n_{i,j}+\Delta t[(\theta_3 k^3_{i,j}T^{n+1}_{i,j+1}+\theta_4 k^4_{i,j}T^{n+1}_{i,j-1})/(\Delta y_j)^2)\ +(2\theta_2 k^2_{i,j}T^{n+1}_{i-1,j} \qquad (51)$$

$$+2(1-\theta_2)k^2_{i,j}T^n_{i-1,j})/(\Delta x_i)^2\ +[(1-\theta_3)k^3_{i,j}(T^n_{i,j+1}-T^n_{i,j})+(1-\theta_4)k^4_{i,j}(T^n_{i,j-1}-T^n_{i,j})]/(\Delta y_j)^2$$

$$-T^{n}_{i,j}((1-\theta_1)k^1_{i,j}+(1-\theta_2)k^2_{i,j})/(\Delta x_i)^2-h_{b4}((2-\theta_1-\theta_2)T^{n}_{i,j}-2T_{b4})/\Delta x_i]$$

Although the last two equations were given for heat transfer at the far left and far right sides of the deicer pad, respectively, (because of the use of the subscipts 3 and 4), they are equally valid for heat transfer at the left and right sides of a variable ice shape if the subscript 2 is used instead. The different heat transfer coefficients are necessary because the far left and far right sides are often taken to be lines of symmetry (h=0) while this is not true for heat transfer at the sides of a variable ice shape. Also, when dealing with variable ice shapes, situations involving heat transfer in rectangular corners is encountered. Combinations of the above equations are used for these cases. The equations for corners are easily obtainable from the above equations by noting that incorporating heat transfer from the top surface does not change in any way the x-derivative terms nor does x-direction heat transfer change y-derivative terms. Hence, the equation for heat transfer at a corner formed by the right and top surfaces is

$$H^{n+1}_{i,j}+\Delta t\, T^{n+1}_{i,j}((\theta_1 k^1_{i,j}+\theta_2 k^2_{i,j})/(\Delta x_i)^2+(\theta_3 k^3_{i,j}+\theta_4 k^4_{i,j})/(\Delta y_j)^2+h_{b2}(\theta_3+\theta_4)/\Delta y_j$$

$$+h_{b4}(\theta_1+\theta_2)/\Delta x_i)=H^{n}_{i,j}+\Delta t[(2\theta_4 k^4_{i,j}T^{n+1}_{i,j-1}+2(1-\theta_4)k^4_{i,j}T^{n}_{i,j-1})/(\Delta y_j)^2$$

$$+(2\theta_2 k^2_{i,j}T^{n+1}_{i-1,j}+2(1-\theta_2)k^2_{i,j}T^{n}_{i-1,j})/(\Delta x_i)^2 \tag{52}$$

$$-[((1-\theta_3)k^3_{i,j}+(1-\theta_4)k^4_{i,j})/(\Delta y_j)^2+((1-\theta_1)k^1_{i,j}+(1-\theta_2)k^2_{i,j})/(\Delta x_i)^2]T^{n}_{i,j}$$

$$-h_{b4}((2-\theta_1-\theta_2)T^{n}_{i,j}-2T_{b4})/\Delta x_i-h_{b2}((2-\theta_3-\theta_4)T^{n}_{i,j}-2T_{b2})/\Delta y_j]$$

Similarly, for heat transfer at a corner formed by the left and top surfaces,

$$H^{n+1}_{i,j}+\Delta t\, T^{n+1}_{i,j}((\theta_1 k^1_{i,j}+\theta_2 k^2_{i,j})/(\Delta x_i)^2+(\theta_3 k^3_{i,j}+\theta_4 k^4_{i,j})/(\Delta y_j)^2+h_{b2}(\theta_3+\theta_4)/\Delta y_j$$

$$+h_{b3}(\theta_1+\theta_2)/\Delta x_i)=H^{n}_{i,j}+\Delta t[(2\theta_4 k^4_{i,j}T^{n+1}_{i,j-1}+2(1-\theta_4)k^4_{i,j}T^{n}_{i,j-1})/(\Delta y_j)^2$$

$$+(2\theta_1 k^1_{i,j}T^{n+1}_{i+1,j}+2(1-\theta_1)k^1_{i,j}T^{n}_{i+1,j})/(\Delta x_i)^2 \tag{53}$$

$$-[((1-\theta_3)k^3{}_{i,j}+(1-\theta_4)k^4{}_{i,j})/\ (\Delta y_j)^2+((1-\theta_1)k^1{}_{i,j}+(1-\theta_2)k^2{}_{i,j})/\ (\Delta x_i)^2\ ]\ T^n{}_{i,j}$$

$$-h_{b3}((2-\theta_1-\theta_2)T^n{}_{i,j}-2T_{b3})/\Delta x_i-h_{b2}((2-\theta_3-\theta_4)T^n{}_{i,j}-2T_{b2})/\Delta y_j]$$

Again, if side heat transfer is from a variable ice shape, h_{b4}, h_{b3}, T_{b3}, and T_{b4} are replaced with h_{b2} and T_{b2}, respectively. It is possible for the variable ice algorithm to create a node which has heat transfer on the left and right surfaces. For this case, a slightly different formulation of the boundary conditions is needed for this substitution process to work. Two one-sided differences are used which, when combined, conserve second order accuracy for all methods. The one-sided differences used are

$$\begin{aligned}&k_{i,j}(\theta_1 T^{n+1}{}_{i+1,j}-\theta_1 T^{n+1}{}_{i,j}+(1-\theta_1)T^n{}_{i+1,j}-(1-\theta_1)T^n{}_{i,j}\ )/2\Delta x_i\\&\qquad=-h_{b2}(2\theta_1 T^{n+1}{}_{i,j}+2(1-\theta_1)T^n{}_{i,j}-2T_{b2})/2\end{aligned}\tag{54a}$$

$$\begin{aligned}&k_{i,j}(\theta_2 T^{n+1}{}_{i-1,j}-\theta_2 T^{n+1}{}_{i,j}+(1-\theta_2)T^n{}_{i-1,j}-(1-\theta_2)T^n{}_{i,j}\ )/2\Delta x_i\\&\qquad=-h_{b2}(2\theta_2 T^{n+1}{}_{i,j}+2(1-\theta_2)T^n{}_{i,j}-2T_{b2})/2\end{aligned}\tag{54b}$$

Adding these two equations produces

$$\begin{aligned}&k_{i,j}(\theta_1 T^{n+1}{}_{i+1,j}+\theta_2 T^{n+1}{}_{i-1,j}-(\theta_1+\theta_2)T^{n+1}{}_{i,j}+(1-\theta_1)T^n{}_{i+1,j}+(1-\theta_2)T^n{}_{i-1,j}\\&-(2-\theta_1-\theta_2)\ T^n{}_{i,j}\ )/2\Delta x_i=-h_{b2}((\theta_1+\theta_2)T^{n+1}{}_{i,j}+(2-\theta_1-\theta_2)T^n{}_{i,j}-2T_{b2})\quad,\end{aligned}\tag{55}$$

or, using average conductivities for the ice layer, one obtains

$$\begin{aligned}&(k^1{}_{i,j}\theta_1 T^{n+1}{}_{i+1,j}+k^2{}_{i,j}\theta_2 T^{n+1}{}_{i-1,j}-(k^1{}_{i,j}\theta_1+k^2{}_{i,j}\theta_2)T^{n+1}{}_{i,j}\\&+k^1{}_{i,j}(1-\theta_1)T^n{}_{i+1,j}+k^2{}_{i,j}(1-\theta_2)T^n{}_{i-1,j}-(k^1{}_{i,j}(1-\theta_1)+k^2{}_{i,j}(1-\theta_2))T^n{}_{i,j})/2(\Delta x_i)^2\\&=-h_{b2}((\theta_1+\theta_2)T^{n+1}{}_{i,j}+(2-\theta_1-\theta_2)T^n{}_{i,j}-2T_{b2})/\Delta x_i\end{aligned}\tag{56}$$

The left hand side of this equation is identical to the finite differenced form of the x-derivative in the heat conduction equation. The right hand side can now be used in the finite differenced form of the conduction equation yielding

$$H^{n+1}_{i,j}+\Delta t\ T^{n+1}_{i,j}(\theta_3 k^3_{i,j}+\theta_4 k^4_{i,j})/(\Delta y_j)^2+2h_{b2}(\theta_1+\theta_2)/\Delta x_i)$$

$$=H^n_{i,j}+\Delta t\ [(\theta_3 k^3_{i,j} T^{n+1}_{i,j+1}+\theta_4 k^4_{i,j} T^{n+1}_{i,j-1})/(\Delta y_j)^2 \tag{57}$$

$$+[(1-\theta_3)k^3_{i,j}(T^n_{i,j+1}-T^n_{i,j})+(1-\theta_4)k^4_{i,j}(T^n_{i,j-1}-T^n_{i,j})]/(\Delta y_j)^2\ -2h_{b2}((2-\theta_1-\theta_2)T^n_{i,j}-2T_{b2})/\Delta x_i$$

For heat transfer on the top, left, and right surfaces:

$$H^{n+1}_{i,j}+\Delta t\ T^{n+1}_{i,j}(\theta_3 k^3_{i,j}+\theta_4 k^4_{i,j})/(\Delta y_j)^2+2h_{b2}(\theta_1+\theta_2)/\Delta x_i\ +h_{b2}(\theta_3+\theta_4)/\Delta y_j))$$

$$=H^n_{i,j}+\Delta t\ [(2\theta_4 k^4_{i,j} T^{n+1}_{i,j-1}+2(1-\theta_4)k^4_{i,j}(T^n_{i,j-1}-T^n_{i,j}))/(\Delta y_j)^2 \tag{58}$$

$$-2h_{b2}((2-\theta_1-\theta_2)T^n_{i,j}-2T_{b2})/\Delta x_i-h_{b2}((2-\theta_3-\theta_4)T^n_{i,j}-2T_{b2})/\Delta y_j]$$

At the interface between the ice layer and the top layer of the composite body, fictitious temperatures are created by assuming that layers are separate entities which are connected to each other through the use of the appropriate boundary condition. Since the ice layer is on top of the blade, this means that the temperatures $T^{n+1}_{i,j-1}$ and $T^n_{i,j-1}$ are not present in the ice layer at this interface, and hence, are considered to be fictitious in the ice layer equation only. Similarly, the temperatures $T^{n+1}_{i,j+1}$ and $T^n_{i,j+1}$ are considered to be fictitious in the energy equation for the composite blade at this interface. This process is also used for interfaces between layers in the composite blade. First, the equations are written to identify the fictitious temperatures in each equation. For the ice layer,

$$(H^{n+1}_{i,j}-H^n_{i,j})/\Delta t=[\theta_1 k^1_{i,j}(T^{n+1}_{i+1,j}-T^{n+1}_{i,j})+\theta_2 k^2_{i,j}(T^{n+1}_{i-1,j}-T^{n+1}_{i,j})]\ /(\Delta x_i)^2$$

$$+[\theta_3 k^3_{i,j}(T^{n+1}_{i,j+1}-T^{n+1}_{i,j})+\theta_4 k^4_{i,j}(T^{n+1}_{i,j-1}{}^{\Delta}-T^{n+1}_{i,j})]/(\Delta y_{j2})^2$$

$$
\begin{aligned}
&+[(1-\theta_1)k^1{}_{i,j}(T^{n}{}_{i+1,j}-T^{n}{}_{i,j})+(1-\theta_2)k^2{}_{i,j}(T^{n}{}_{i-1,j}-T^{n}{}_{i,j})]/(\Delta x_i)^2 \\
&+[(1-\theta_3)k^3{}_{i,j}(T^{n}{}_{i,j+1}-T^{n}{}_{i,j})+(1-\theta_4)k^4{}_{i,j}(T^{n}{}_{i,j-1}{}^{\Delta}-T^{n}{}_{i,j})]/(\Delta y_{j2})^2
\end{aligned}
\tag{59}
$$

For the composite body,

$$
\begin{aligned}
&(T^{n+1}{}_{i,j}-T^{n}{}_{i,j})/\Delta t=\alpha_{i,j1}[\theta_1(T^{n+1}{}_{i+1,j}-T^{n+1}{}_{i,j})+\theta_2(T^{n+1}{}_{i-1,j}-T^{n+1}{}_{i,j})]/(\Delta x_i)^2 \\
&+\alpha_{i,j1}[\theta_3(T^{n+1}{}_{i,j+1}{}^{\Delta}-T^{n+1}{}_{i,j})+\theta_4(T^{n+1}{}_{i,j-1}-T^{n+1}{}_{i,j})]/(\Delta y_{j1})^2 \\
&+\alpha_{i,j1}[(1-\theta_1)(T^{n}{}_{i+1,j}-T^{n}{}_{i,j})+(1-\theta_2)(T^{n}{}_{i-1,j}-T^{n}{}_{i,j})]/(\Delta x_i)^2 \\
&+\alpha_{i,j1}[(1-\theta_3)(T^{n}{}_{i,j+1}{}^{\Delta}-T^{n}{}_{i,j})+(1-\theta_4)(T^{n}{}_{i,j-1}-T^{n}{}_{i,j})]/(\Delta y_j)^2
\end{aligned}
\tag{60}
$$

The y-direction boundary condition is

$$
\begin{aligned}
&-k_{i,j1}(\theta_3T^{n+1}{}_{i,j+1}{}^{\Delta}-\theta_4T^{n+1}{}_{i,j-1})/4\Delta y_{j1}-k_{i,j1}((1-\theta_3)T^{n}{}_{i,j+1}{}^{\Delta}-(1-\theta_4)T^{n}{}_{i,j-1})/4\Delta y_{j1} \\
&=-k_{i,j2}(\theta_3T^{n+1}{}_{i,j+1}-\theta_4T^{n+1}{}_{i,j-1}{}^{\Delta})/4\Delta y_{j2}-k_{i,j2}((1-\theta_3)T^{n}{}_{i,j+1}-(1-\theta_4)T^{n}{}_{i,j-1}{}^{\Delta})/4\Delta y_{j2}
\end{aligned}
\tag{61}
$$

The procedure for eliminating the fictitious temperatures is as follows: (1) solve the boundary condition equation for $T^{n+1}{}_{i,j-1}{}^{\Delta}$ and $T^{n}{}_{i,j-1}{}^{\Delta}$; (2) substitute these temperatures into the ice layer equation; (3) solve the energy equation for the composite blade for the fictitious temperatures $T^{n+1}{}_{i,j+1}{}^{\Delta}$ and $T^{n}{}_{i,j+1}{}^{\Delta}$; (4) eliminate the remaining two fictitious temperatures between the ice layer equation and the composite body equation; and finally, (5) group all terms with the same temperature and solve for $T^{n+1}{}_{i,j}$. These steps are performed as follows:

Step 1:

$$
\begin{aligned}
&-(\theta_3T^{n+1}{}_{i,j+1}{}^{\Delta}+(1-\theta_3)T^{n}{}_{i,j+1}{}^{\Delta}-\theta_4T^{n+1}{}_{i,j-1}-(1-\theta_4)T^{n}{}_{i,j-1})(k_{i,j1}\Delta y_{j2}) \\
&/(k_{i,j2}\Delta y_{j1})+\theta_3T^{n+1}{}_{i,j+1}+(1-\theta_3)T^{n}{}_{i,j+1}=\theta_4T^{n+1}{}_{i,j-1}{}^{\Delta}+(1-\theta_4)T^{n}{}_{i,j-1}{}^{\Delta}
\end{aligned}
\tag{62}
$$

Step 2: substitute Eq. (62) into the ice layer equation to produce

$$\begin{aligned}
&(H^{n+1}_{i,j}-H^{n}_{i,j})/\Delta t=[\theta_1 k^1_{i,j}(T^{n+1}_{i+1,j}-T^{n+1}_{i,j})+\theta_2 k^2_{i,j}(T^{n+1}_{i-1,j}-T^{n+1}_{i,j})]\,/(\Delta x_i)^2\\
&+[\theta_3 k^3_{i,j}(T^{n+1}_{i,j+1}-T^{n+1}_{i,j})-\theta_4 k^4_{i,j}(T^{n+1}_{i,j})]/(\Delta y_{j2})^2\\
&+[(1-\theta_1)k^1_{i,j}(T^{n}_{i+1,j}-T^{n}_{i,j})+(1-\theta_2)k^2_{i,j}(\,T^{n}_{i-1,j}-T^{n}_{i,j})]/(\Delta x_i)^2\\
&+[(1-\theta_3)k^3_{i,j}(T^{n}_{i,j+1}-T^{n}_{i,j})-(1-\theta_4)k^4_{i,j}(T^{n}_{i,j})]/(\Delta y_{j2})^2\\
&+k^4_{i,j}\,/(\Delta y_{j2})^2\,(\,-(\theta_3 {T^{n+1}_{i,j+1}}^{\Delta}+(1-\theta_3){T^{n}_{i,j+1}}^{\Delta}-\theta_4 T^{n+1}_{i,j-1}\\
&-(1-\theta_4)T^{n}_{i,j-1}\,)\,(k_{i,j1}\Delta y_{j2})/(k_{i,j2}\,\Delta y_{j1})+\theta_3 T^{n+1}_{i,j+1}+(1-\theta_3)T^{n}_{i,j+1})
\end{aligned} \tag{63}$$

Step 3: solve the energy equation for the composite blade for the last two fictitious temperatures

$$\begin{aligned}
&(\Delta y_{j1})^2(T^{n+1}_{i,j}-T^{n}_{i,j})/(\alpha_{i,j1}\,\Delta t)-[\theta_1(T^{n+1}_{i+1,j}-T^{n+1}_{i,j})\\
&+\theta_2(T^{n+1}_{i-1,j}-T^{n+1}_{i,j})]\,(\Delta y_{j1})^2/(\Delta x_i)^2\,-\theta_4(T^{n+1}_{i,j-1}-T^{n+1}_{i,j})\\
&+\theta_3 T^{n+1}_{i,j}+(2-\theta_3-\theta_4)T^{n}_{i,j}-(1-\theta_4)T^{n}_{i,j-1}-[(1-\theta_1)(T^{n}_{i+1,j}-T^{n}_{i,j})\\
&+(1-\theta_2)(T^{n}_{i-1,j}-T^{n}_{i,j})]\,(\Delta y_{j1})^2/(\Delta x_i)^2=(1-\theta_3){T^{n}_{i,j+1}}^{\Delta}+\theta_3\,{T^{n+1}_{i,j+1}}^{\Delta}
\end{aligned} \tag{64}$$

Step 4: eliminate the fictitious temperatures between Eqs.(63) and (64) to get

$$\begin{aligned}
&(H^{n+1}_{i,j}-H^{n}_{i,j})/\Delta t=[\theta_1 k^1_{i,j}(T^{n+1}_{i+1,j}-T^{n+1}_{i,j})+\theta_2 k^2_{i,j}(T^{n+1}_{i-1,j}-T^{n+1}_{i,j})]\,/(\Delta x_i)^2\\
&+[\theta_3 k^3_{i,j}(T^{n+1}_{i,j+1}-T^{n+1}_{i,j})-\theta_4 k^4_{i,j}(T^{n+1}_{i,j})]/(\Delta y_{j2})^2\\
&+[(1-\theta_1)k^1_{i,j}(T^{n}_{i+1,j}-T^{n}_{i,j})+(1-\theta_2)k^2_{i,j}(\,T^{n}_{i-1,j}-T^{n}_{i,j})]/(\Delta x_i)^2\\
&+[(1-\theta_3)k^3_{i,j}(T^{n}_{i,j+1}-T^{n}_{i,j})-(1-\theta_4)k^4_{i,j}(T^{n}_{i,j})]/(\Delta y_{j2})^2\\
&+k^4_{i,j}\,/(\Delta y_{j2})^2\,(\,-\theta_4 T^{n+1}_{i,j-1}-(1-\theta_4)T^{n}_{i,j-1}\,)\,(k_{i,j1}\Delta y_{j2})/(k_{i,j2}\Delta y_{j1})
\end{aligned} \tag{65}$$

$$
\begin{aligned}
&+\theta_3 T^{n+1}_{i,j+1}+(1-\theta_3)T^{n}_{i,j+1}) +(k^{4}_{i,j}\, k_{i,j1}) /(k_{i,j2}\, \Delta y_{j2}\, \Delta y_{j1})\\
&[(\Delta y_{j1})^2(T^{n+1}_{i,j}-T^{n}_{i,j})/(\alpha_{i,j1}\, \Delta t)-[\theta_1(T^{n+1}_{i+1,j}-T^{n+1}_{i,j})\\
&+\theta_2(T^{n+1}_{i-1,j}-T^{n+1}_{i,j})]\,(\Delta y_{j1})^2/(\Delta x_i)^2-\theta_4(T^{n+1}_{i,j-1}-T^{n+1}_{i,j}) +\theta_3 T^{n+1}_{i,j}\\
&+(2-\theta_3-\theta_4)\, T^{n}_{i,j} -(1-\theta_4)T^{n}_{i,j-1}-[(1-\theta_1)(T^{n}_{i+1,j}-T^{n}_{i,j})+(1-\theta_2)(T^{n}_{i-1,j}-T^{n}_{i,j})]\,(\Delta y_{j1})^2/(\Delta x_i)^2]
\end{aligned}
$$

Step 5: Finally, group like terms and solve for $T^{n+1}_{i,j}$ which yields

$$
\begin{aligned}
&H^{n+1}_{i,j}/\Delta t +[(\theta_1 k^{1}_{i,j} +\theta_2 k^{2}_{i,j})/(\Delta x_i)^2+(\theta_3 k^{3}_{i,j}+\theta_4 k^{4}_{i,j})/(\Delta y_{j2})^2\\
&+(k^{4}_{i,j}\, k_{i,j1})/(k_{i,j2}\Delta y_{j2}\Delta y_{j1})[(\Delta y_{j1})^2/(\alpha_{i,j1}\Delta t) +(\Delta y_{j1})^2/(\Delta x_i)^2(\theta_1+ \theta_2)+\theta_3+\theta_4]]\; T^{n+1}_{i,j}\\
&=H^{n}_{i,j}/\Delta t+[\theta_1 k^{1}_{i,j}T^{n+1}_{i+1,j}+\theta_2 k^{2}_{i,j}T^{n+1}_{i-1,j}]/(\Delta x_i)^2+(k^{4}_{i,j}k_{i,j1}\Delta y_{j1})\\
&/(k_{i,j2}\Delta y_{j2}(\Delta x_i)^2)[\theta_1 T^{n+1}_{i+1,j}+\theta_2 T^{n+1}_{i-1,j}]+2(\theta_3 k^{3}_{i,j}T^{n+1}_{i,j+1}\\
&+(1-\theta_3)k^{3}_{i,j}T^{n}_{i,j+1})+(k_{i,j1}k^{4}_{i,j}/(k_{i,j2}\Delta y_{j1}\Delta y_{j2})\,(\theta_4 T^{n+1}_{i,j-1}+(1-\theta_4)T^{n}_{i,j-1})\\
&+[(1-\theta_1)k^{1}_{i,j}(T^{n}_{i+1,j}-T^{n}_{i,j})(1-\theta_2)k^{2}_{i,j}(T^{n}_{i-1,j}-T^{n}_{i,j})]/(\Delta x_i)^2+[(1-\theta_3)k^{3}_{i,j}(T^{n}_{i,j+1}-T^{n}_{i,j})\\
&-(1-\theta_4)k^{4}_{i,j}(T^{n}_{i,j})]/(\Delta y_{j2})^2 +(k^{4}_{i,j}\, k_{i,j1}) /(k_{i,j2}\, \Delta y_{j2}\, \Delta y_{j1})[-\Delta y^{2}_{j1}T^{n}_{i,j})/(\alpha_{i,j1}\, \Delta t)\\
&+(2-\theta_3-\theta_4)T^{n}_{i,j}-(1-\theta_4)T^{n}_{i,j-1}-[(1-\theta_1)(T^{n}_{i+1,j} -T^{n}_{i,j}) +(1-\theta_2)(T^{n}_{i-1,j} -T^{n}_{i,j})](\Delta y_j)^2/(\Delta x_i)^2]
\end{aligned}
\tag{66}
$$

For side heat transfer at an interface, the same analysis is performed as was performed for side heat transfer alone. For heat transfer on the left side, Eq. (46a) is solved for the fictitious temperatures $T^{n+1}_{i-1,j}$ and $T^{n}_{i-1,j}$ and then substituted into Eq. (66) to get

$$
\begin{aligned}
&H^{n+1}_{i,j}/\Delta t +[(\theta_1 k^{1}_{i,j} +\theta_2 k^{2}_{i,j})/(\Delta x_i)^2+(\theta_3 k^{3}_{i,j}+\theta_4 k^{4}_{i,j})/(\Delta y_{j2})^2\\
&+(k^{4}_{i,j}\, k_{i,j1})/(k_{i,j2}\Delta y_{j2}\Delta y_{j1})[(\Delta y_{j1})^2/(\alpha_{i,j1}\, \Delta t)+(\Delta y_{j1})^2/(\Delta x_i)^2(\theta_1+\theta_2)+\theta_3+\theta_4]
\end{aligned}
$$

$$\begin{aligned}
&+h_{b3}(\theta_1 k^{1}_{i,j}/(k_{i,j2}\Delta y_{j2}(\Delta x_i)^2)+(k^{4}_{i,j}k_{i,j1}\Delta y_{j1})/(k_{i,j2}\Delta y_{j2}(\Delta x_i)^2))]T^{n+1}_{i,j}\\
&=H^{n}_{i,j}/\Delta t+[2\theta_1 k^{1}_{i,j}T^{n+1}_{i+1,j}]/(\Delta x_i)^2+2(k^{4}_{i,j}k_{i,j1}\Delta y_{j1})/(k_{i,j2}\Delta y_{j2}(\Delta x_i)^2)[\theta_1 T^{n+1}_{i+1,j}]\\
&+2(\theta_3 k^{3}_{i,j}T^{n+1}_{i,j+1}+(1-\theta_3)k^{3}_{i,j}T^{n}_{i,j+1})\\
&+(k_{i,j1}k^{4}_{i,j}/(k_{i,j2}\Delta y_{j1}\Delta y_{j2})(\theta_4 T^{n+1}_{i,j-1}+(1-\theta_4)T^{n}_{i,j-1})\\
&+[(1-\theta_1)k^{1}_{i,j}(2T^{n}_{i+1,j}-T^{n}_{i,j})-(1-\theta_2)k^{2}_{i,j}(T^{n}_{i,j})]/(\Delta x_i)^2\\
&+[(1-\theta_3)k^{3}_{i,j}(T^{n}_{i,j+1}-T^{n}_{i,j})-(1-\theta_4)k^{4}_{i,j}(T^{n}_{i,j})]/(\Delta y_{j2})^2\\
&+(k^{4}_{i,j}k_{i,j1})/(k_{i,j2}\Delta y_{j2}\Delta y_{j1})[-(\Delta y_{j1})^2T^{n}_{i,j})/(\alpha_{i,j1}\Delta t)\\
&+(2-\theta_3-\theta_4)T^{n}_{i,j}-(1-\theta_4)T^{n}_{i,j-1}-[(1-\theta_1)(2T^{n}_{i+1,j}-T^{n}_{i,j})\\
&-(1-\theta_2)(T^{n}_{i,j})](\Delta y_{j1}/\Delta x_i)^2]-(\theta_1 k^{1}_{i,j}/(\Delta x_i)^2+(k^{4}_{i,j}k_{i,j1}\Delta y_{j1})/(k_{i,j2}\Delta y_{j2}(\Delta x_i)^2))\\
&h_{b3}[(2-\theta_3-\theta_4)T^{n}_{i,j}-2T_{b3}]/\Delta y_{j2}
\end{aligned}\tag{67}$$

Similarly, for heat transfer at the right side of this interface,

$$\begin{aligned}
&H^{n+1}_{i,j}/\Delta t+[(\theta_1 k^{1}_{i,j}+\theta_2 k^{2}_{i,j})/(\Delta x_i)^2+(\theta_3 k^{3}_{i,j}+\theta_4 k^{4}_{i,j})/(\Delta y_{j2})^2\\
&+(k^{4}_{i,j}k_{i,j1})/(k_{i,j2}\Delta y_{j2}\Delta y_{j1})[(\Delta y_{j1})^2/(\alpha_{i,j1}\Delta t)+(\Delta y_{j1}/\Delta x_i)^2(\theta_1+\theta_2)+\theta_3+\theta_4]\\
&+h_{b3}(\theta_1 k^{1}_{i,j}/(k_{i,j2}\Delta y_{j2}(\Delta x_i)^2)+(k^{4}_{i,j}k_{i,j1}\Delta y_{j1})/(k_{i,j2}\Delta y_{j2}(\Delta x_i)^2))]T^{n+1}_{i,j}\\
&=H^{n}_{i,j}/\Delta t+[2\theta_1 k^{1}_{i,j}T^{n+1}_{i+1,j}]/(\Delta x_i)^2+2(k^{4}_{i,j}k_{i,j1}\Delta y_{j1})/(k_{i,j2}\Delta y_{j2}(\Delta x_i)^2)\\
&[\theta_1 T^{n+1}_{i+1,j}]+2(\theta_3 k^{3}_{i,j}T^{n+1}_{i,j+1}+(1-\theta_3)k^{3}_{i,j}T^{n}_{i,j+1})\\
&+(k_{i,j1}k^{4}_{i,j}/(k_{i,j2}\Delta y_{j1}\Delta y_{j2})(\theta_4 T^{n+1}_{i,j-1}+(1-\theta_4)T^{n}_{i,j-1})\\
&+[(1-\theta_1)k^{1}_{i,j}(2T^{n}_{i+1,j}-T^{n}_{i,j})-(1-\theta_2)k^{2}_{i,j}(T^{n}_{i,j})]/(\Delta x_i)^2
\end{aligned}\tag{68}$$

$$+[(1-\theta_3)k^3_{i,j}(T^n_{i,j+1}-T^n_{i,j})-(1-\theta_4)k^4_{i,j}(T^n_{i,j})]/(\Delta y_{j2})^2$$

$$+(k^4_{i,j}\,k_{i,j1})/(k_{i,j2}\,\Delta y_{j2}\,\Delta y_{j1})[-(\Delta y_{j1})^2T^n_{i,j})/(\alpha_{i,j1}\Delta t)$$

$$+(2-\theta_3-\theta_4)T^n_{i,j}-(1-\theta_4)T^n_{i,j-1}-[(1-\theta_1)(2T^n_{i+1,j}-T^n_{i,j})$$

$$-(1-\theta_2)(T^n_{i,j})](\Delta y_{j1}/\Delta x_i)^2]-(\theta_1k^1_{i,j}/(\Delta x_i)^2+(k^4_{i,j}k_{i,j1}\Delta y_{j1})/(k_{i,j2}\Delta y_{j2}(\Delta x_i)^2)$$

$$h_{b3}[(2-\theta_3-\theta_4)T^n_{i,j}-2T_{b3}]/\Delta y_{j2}$$

So far, the equations have dealt exclusively with the manner in which the boundaries interact with the energy equation for the ice layer through the application of boundary conditions. For the remainder of the composite body, the same boundaries can occur, and hence, the same derivations can be performed for the composite body. Since the energy equation for the ice layer is so similar to that for the composite body, the boundary condition equations derived previously will not be rederived for the composite body.

The differences between the composite body equations and the ice layer equations are: (1) no average thermal conductivities are needed since no phase change occurs; (2) enthalpy is expressed in terms of temperature using $T_{i,j}=H_{i,j}/\rho_{i,j}C_{p\,i,j}$, because no phase change takes place in the composite body; and (3) a heat source term is present in the composite body in order to account for the heater layer. The form of this term will become apparant in the derivations which follow.

The only boundary condition equations for the composite body which will be derived here will be those which do not occur in the ice layer. These pertain to heat transfer at the bottom surface and x-direction interfaces between layers. The latter occurs only in the heater layer. For the heat transfer at the bottom of the composite body, Eq. (46d) is first solved for the fictitious temperatures $T^{n+1}_{i,j-1}$ and $T^n_{i,j-1}$:

$$\theta_4T^{n+1}_{i,j-1}{}^{\Delta}+(1-\theta_4)T^n_{i,j-1}{}^{\Delta}=\theta_3T^{n+1}_{i,j+1}+(1-\theta_3)T^n_{i,j+1}$$

$$-\Delta y_jh_{b1}((\theta_3+\theta_4)T^{n+1}_{i,j}+(2-\theta_3-\theta_4)T^n_{i,j}-2T_{b1})/k_{i,j} \qquad (69)$$

Next the fictitious temperatures can be eliminated by substitution into Eq. (42) to obtain

$$
\begin{aligned}
(T^{n+1}_{i,j}-T^{n}_{i,j})/\Delta t=&\alpha_{i,j}[\theta_1(T^{n+1}_{i+1,j}-T^{n+1}_{i,j})+\theta_2(T^{n+1}_{i-1,j}-T^{n+1}_{i,j})]/(\Delta x_i)^2\\
&+\alpha_{i,j}[\theta_3(2T^{n+1}_{i,j+1}-T^{n+1}_{i,j})-\theta_4T^{n+1}_{i,j}]/(\Delta y_j)^2\\
&+\alpha_{i,j}[(1-\theta_1)(T^{n}_{i+1,j}-T^{n}_{i,j})+(1-\theta_2)(T^{n}_{i-1,j}-T^{n}_{i,j})]/(\Delta x_i)^2\\
&+\alpha_{i,j}[(1-\theta_3)(2T^{n}_{i,j+1}-T^{n}_{i,j})-(1-\theta_4)T^{n}_{i,j}]/(\Delta y_j)^2\\
&-\alpha_{i,j}/(\Delta y_j)^2(\Delta y_j h_{b1}((\theta_3+\theta_4)T^{n+1}_{i,j}+(2-\theta_3-\theta_4)T^{n}_{i,j}-2T_{b1})/k_{i,j})
\end{aligned}
\tag{70}
$$

Rearranging in order to solve for $T^{n+1}_{i,j}$ produces

$$
\begin{aligned}
&T^{n+1}_{i,j}[1/\Delta t+\alpha_{i,j}((\theta_1+\theta_2)/(\Delta x_i)^2+(\theta_3+\theta_4)/(\Delta y_j)^2+h_{b1}(\theta_3+\theta_4)/(k_{i,j}\Delta y_j))]\\
&=\alpha_{i,j}(\theta_1T^{n+1}_{i+1,j}+(1-\theta_1)T^{n}_{i+1,j}+\theta_2T^{n+1}_{i-1,j}+(1-\theta_2)T^{n}_{i-1,j})/(\Delta x_i)^2\\
&+2\alpha_{i,j}(\theta_3T^{n+1}_{i,j+1}+(1-\theta_3)T^{n}_{i,j+1})/(\Delta y_j)^2-T^{n}_{i,j}[1/\Delta t+\alpha_{i,j}((2-\theta_1-\theta_2)/(\Delta x_i)^2\\
&+(2-\theta_3-\theta_4)/(\Delta y_j)^2+h_{b1}(2-\theta_3-\theta_4)/(k_{i,j}\Delta y_j))]+2\alpha_{i,j}h_{b1}T_{b1}/(k_{i,j}\Delta y_j)
\end{aligned}
\tag{71}
$$

For heat transfer on the left and bottom corner of the grid, Eq. (46c) is solved for the fictitious temperatures $T^{n+1}_{i-1,j}$ and $T^{n}_{i-1,j}$. Then it is substituted into the above equation which is then solved for $T^{n+1}_{i,j}$:

$$
\begin{aligned}
\theta_2T^{n+1}_{i-1,j}{}^{\Delta}+(1-\theta_2)T^{n}_{i-1,j}{}^{\Delta}=&\theta_1T^{n+1}_{i+1,j}+(1-\theta_1)T^{n}_{i+1,j}\\
&-\Delta x_i h_{b3}((\theta_1+\theta_2)T^{n+1}_{i,j}+(2-\theta_1-\theta_2)T^{n}_{i,j}-2T_{b3})/k_{i,j}
\end{aligned}
\tag{72}
$$

Now substitute Eq. (72) into Eq. (71) to get

$$
\begin{aligned}
&T^{n+1}_{i,j}[1/\Delta t+\alpha_{i,j}((\theta_1+\theta_2)/(\Delta x_i)^2+(\theta_3+\theta_4)/(\Delta y_j)^2+h_{b1}(\theta_3+\theta_4)/(k_{i,j}\Delta y_j)\\
&+h_{b3}(\theta_1+\theta_2)/(k_{i,j}\Delta x_i))]=2\alpha_{i,j}(\theta_1T^{n+1}_{i+1,j}+(1-\theta_1)T^{n}_{i+1,j})/(\Delta x_i)^2
\end{aligned}
$$

$$+2\alpha_{i,j}(\theta_3 T^{n+1}_{i,j+1}+(1-\theta_3)T^{n}_{i,j+1})/(\Delta y_j)-T^{n}_{i,j}[1/\Delta t+\alpha_{i,j}((2-\theta_1-\theta_2)/(\Delta x_i)^2 \tag{73}$$

$$+(2-\theta_3-\theta_4)/(\Delta y_j)^2+h_{b1}(2-\theta_3-\theta_4)/(k_{i,j}\Delta y_j)+h_{b3}(\theta_1+\theta_2)/(k_{i,j}\Delta x_i))]$$

$$+2\alpha_{i,j}h_{b1}T_{b1}/(k_{i,j}\Delta y_j)+2\alpha_{i,j}h_{b3}T_{b3}/(k_{i,j}\Delta x_i)$$

Similarly, for the bottom right corner, the equation is

$$T^{n+1}_{i,j}[1/\Delta t+\alpha_{i,j}((\theta_1+\theta_2)/(\Delta x_i)^2+(\theta_3+\theta_4)/(\Delta y_j)^2+h_{b1}(\theta_3+\theta_4)/(k_{i,j}\Delta y_j)$$

$$+h_{b4}(\theta_1+\theta_2)/(k_{i,j}\Delta x_i))]=2\alpha_{i,j}(\theta_2 T^{n+1}_{i-1,j}+(1-\theta_2)T^{n}_{i-1,j})/(\Delta x_i)^2$$

$$+2\alpha_{i,j}(\theta_3 T^{n+1}_{i,j+1}+(1-\theta_3)T^{n}_{i,j+1})/(\Delta y_j)^2-T^{n}_{i,j}[1/\Delta t+\alpha_{i,j}((2-\theta_1-\theta_2)/(\Delta x_i)^2 \tag{74}$$

$$+(2-\theta_3-\theta_4)/(\Delta y_j)^2+h_{b1}(2-\theta_3-\theta_4)/(k_{i,j}\Delta y_j)+h_{b4}(\theta_1+\theta_2)/(k_{i,j}\Delta x_i))]$$

$$+2\alpha_{i,j}h_{b1}T_{b1}/(k_{i,j}\Delta y_j)+2\alpha_{i,j}h_{b4}T_{b4}/(k_{i,j}\Delta x_i)$$

The above equations do not contain heat source terms because the heater is an interior layer, not located on the bottom boundary. For the heater layer, Eq. (42) is used with a heat source term added. This is represented by

$$(T^{n+1}_{i,j}-T^{n}_{i,j})/\Delta t=\alpha_{i,j}[\theta_1(T^{n+1}_{i+1,j}-T^{n+1}_{i,j})+\theta_2(T^{n+1}_{i-1,j}-T^{n+1}_{i,j})]/(\Delta x_i)^2$$

$$+\alpha_{i,j}[\theta_3(T^{n+1}_{i,j+1}-T^{n+1}_{i,j})+\theta_4(T^{n+1}_{i,j-1}-T^{n+1}_{i,j})]/(\Delta y_j)^2$$

$$+\alpha_{i,j}[(1-\theta_1)(T^{n}_{i+1,j}-T^{n}_{i,j})+(1-\theta_2)(T^{n}_{i-1,j}-T^{n}_{i,j})]/(\Delta x_i)^2 \tag{75}$$

$$+\alpha_{i,j}[(1-\theta_3)(T^{n}_{i,j+1}-T^{n}_{i,j})+(1-\theta_4)(T^{n}_{i,j-1}-T^{n}_{i,j})]/(\Delta y_j)^2$$

$$+q_{i,j}\alpha_{i,j}/(k_{i,j}\Delta y_j)$$

Since in most cases the heater is assumed to be finite in the x-direction, an interface arises between the heater and the insulation surrounding it. The equation for this interface is determined in much the same way as the equation for the interface between the ice layer and the composite body. First, the x-direction interfacial boundary condition equation is written to identify the fictitious temperatures which need to be eliminated. Using Eq. (45a),

$$\begin{aligned}
&-k_{i1,j}(\theta_1 {T^{n+1}_{i+1,j}}^{\Delta}-\theta_2 T^{n+1}_{i-1,j})/4\Delta x_{i1}-k_{i1,j}((1-\theta_1)T^{n}_{i+1,j}{}^{\Delta}\\
&-(1-\theta_2)T^{n}_{i-1,j})/4\Delta x_{i1}=-k_{i2,j}(\theta_1 T^{n+1}_{i+1,j}-\theta_2 {T^{n+1}_{i-1,j}}^{\Delta})/4\Delta x_{i2}\\
&-k_{i2,j}((1-\theta_1)T^{n}_{i+1,j}\ -(1-\theta_2){T^{n}_{i-1,j}}^{\Delta})/4\Delta x_{i2}
\end{aligned}\tag{45a}$$

Solving this for ${T^{n+1}_{i+1,j}}^{\Delta}$ and ${T^{n+1}_{i+1,j}}^{\Delta}$ yields

$$\begin{aligned}
&\theta_1 {T^{n+1}_{i+1,j}}^{\Delta}+(1-\theta_1){T^{n}_{i+1,j}}^{\Delta}=\theta_2 T^{n+1}_{i-1,j}+(1-\theta_2)T^{n}_{i-1,j}+k_{i2,j}\Delta x_{i2}\\
&(\theta_1 T^{n+1}_{i+1,j}+(1-\theta_1)T^{n}_{i+1,j}-(\theta_2 {T^{n+1}_{i-1,j}}^{\Delta}+(1-\theta_2){T^{n}_{i-1,j}}^{\Delta}))/(\Delta x_{i2}k_{i1,j})
\end{aligned}\tag{76}$$

Equation (75) is now written for layer i1:

$$\begin{aligned}
&(T^{n+1}_{i,j}-T^{n}_{i,j})/\Delta t=\alpha_{i1,j}[\theta_1({T^{n+1}_{i+1,j}}^{\Delta}-T^{n+1}_{i,j})+\theta_2(T^{n+1}_{i-1,j}-T^{n+1}_{i,j})]\ /(\Delta x_{i1})^2\\
&\qquad+\alpha_{i1,j}[\theta_3(T^{n+1}_{i,j+1}-T^{n+1}_{i,j})+\theta_4(T^{n+1}_{i,j-1}-T^{n+1}_{i,j})]/(\Delta y_j)^2\\
&\qquad+\alpha_{i1,j}[(1-\theta_1)({T^{n}_{i+1,j}}^{\Delta}-T^{n}_{i,j})+(1-\theta_2)(T^{n}_{i-1,j}-T^{n}_{i,j})]/(\Delta x_{i1})^2\\
&\qquad+\alpha_{i1,j}[(1-\theta_3)(T^{n}_{i,j+1}-T^{n}_{i,j})+(1-\theta_4)(T^{n}_{i,j-1}-T^{n}_{i,j})]/(\Delta y_j)^2\\
&\qquad+q_{i1,j}\alpha_{i1,j}/(k_{i1,j}\Delta y_j)
\end{aligned}\tag{77}$$

Equation (74) is now substituted into Eq. (75) to get

$$\begin{aligned}
&(T^{n+1}_{i,j}-T^{n}_{i,j})/\Delta t=\alpha_{i1,j}[\theta_1(-T^{n+1}_{i,j})+\theta_2(2T^{n+1}_{i-1,j}-T^{n+1}_{i,j})]/(\Delta x_{i1})^2\\
&+\alpha_{i1,j}[\theta_3(T^{n+1}_{i,j+1}-T^{n+1}_{i,j})+\theta_4(T^{n+1}_{i,j-1}-T^{n+1}_{i,j})]/(\Delta y_j)^2\\
&\qquad+\alpha_{i1,j}[(1-\theta_1)(-T^{n}_{i,j})+(1-\theta_2)(2T^{n}_{i-1,j}-T^{n}_{i,j})]/(\Delta x_{i1})^2
\end{aligned}\tag{78}$$

$$+\alpha_{i1,j}[(1-\theta_3)(T^n_{i,j+1}-T^n_{i,j})+(1-\theta_4)(T^n_{i,j-1}-T^n_{i,j})]/(\Delta y_j)^2+q_{i1,j}\alpha_{i1,j}/(k_{i1,j}\Delta y_j)$$

$$+k_{i2,j}\Delta x_{i2}(\theta_1 T^{n+1}_{i+1,j}+(1-\theta_1)T^n_{i+1,j}-(\theta_2 {T^{n+1}_{i-1,j}}^{\Delta}+(1-\theta_2){T^n_{i-1,j}}^{\Delta}))/(\Delta x_{i2}k_{i1,j})$$

Next Eq. (75) is written for layer i2:

$$(T^{n+1}_{i,j}-T^n_{i,j})/\Delta t=\alpha_{i2,j}[\theta_1(T^{n+1}_{i+1,j}-T^{n+1}_{i,j})+\theta_2({T^{n+1}_{i-1,j}}^{\Delta}-T^{n+1}_{i,j})]/(\Delta x_{i2})^2$$

$$+\alpha_{i2,j}[\theta_3(T^{n+1}_{i,j+1}-T^{n+1}_{i,j})+\theta_4(T^{n+1}_{i,j-1}-T^{n+1}_{i,j})]/(\Delta y_j)^2$$

$$+\alpha_{i2,j}[(1-\theta_1)(T^n_{i+1,j}-T^n_{i,j})+(1-\theta_2)({T^n_{i-1,j}}^{\Delta}-T^n_{i,j})]/(\Delta x_{i2})^2 \tag{79}$$

$$+\alpha_{i2,j}[(1-\theta_3)(T^n_{i,j+1}-T^n_{i,j})+(1-\theta_4)(T^n_{i,j-1}-T^n_{i,j})]/(\Delta y_j)^2$$

$$+q_{i2,j}\alpha_{i2,j}/(k_{i2,j}\Delta y_j)$$

Equations (78) and (79) are combined to eliminate ${T^{n+1}_{i-1,j}}^{\Delta}$ and ${T^n_{i-1,j}}^{\Delta}$ to obtain

$$(T^{n+1}_{i,j}-T^n_{i,j})/\Delta t=\alpha_{i1,j}[\theta_1(-T^{n+1}_{i,j})+\theta_2(2T^{n+1}_{i-1,j}-T^{n+1}_{i,j})]/(\Delta x_{i1})^2$$

$$+\alpha_{i1,j}[\theta_3(T^{n+1}_{i,j+1}-T^{n+1}_{i,j})+\theta_4(T^{n+1}_{i,j-1}-T^{n+1}_{i,j})]/(\Delta y_j)^2$$

$$+\alpha_{i1,j}[(1-\theta_1)(-T^n_{i,j})+(1-\theta_2)(2T^n_{i-1,j}-T^n_{i,j})]/(\Delta x_{i1})^2$$

$$+\alpha_{i1,j}[(1-\theta_3)(T^n_{i,j+1}-T^n_{i,j})+(1-\theta_4)(T^n_{i,j-1}-T^n_{i,j})]/(\Delta y_j)^2$$

$$+q_{i1,j}\alpha_{i1,j}/(k_{i1,j}\Delta y_j)+\alpha_{i1,j}k_{i2,j}/(\Delta x_{i1}\Delta x_{i2}k_{i1,j}) \tag{80}$$

$$[2\theta_1 T^{n+1}_{i+1,j}+2(1-\theta_1)T^n_{i+1,j}-(\Delta x_{i2})^2(T^{n+1}_{i,j}-T^n_{i,j})/(\alpha_{i2,j}\,\Delta t)$$

$$-((\theta_1+\theta_2)+(\theta_3+\theta_4)(\Delta x_{i2}/\Delta y_j)^2)T^{n+1}_{i,j}$$

$$+(\theta_3 T^{n+1}_{i,j+1}+(1-\theta_3)T^n_{i,j+1}+\theta_4 T^{n+1}_{i,j-1}+(1-\theta_4)T^n_{i,j-1})(\Delta x_{i2}/\Delta y_j)^2$$

$$-(2-\theta_1-\theta_2+(2-\theta_3-\theta_4)(\Delta x_{i2}/\Delta y_j)^2)T^n_{i,j}+q_{i2,j}\alpha_{i2,j}/(k_{i2,j}\Delta y_j)]$$

Solving the above equation for $T^{n+1}_{i,j}$ produces the boundary condition equation for the heat transfer in the x-direction at the interface between heater and insulation, i.e.,

$$\begin{gathered}
T^{n+1}_{i,j}[1/\Delta t+\alpha_{i1,j}((\theta_1+\theta_2)/(\Delta x_{i1})^2+(\theta_3+\theta_4)/(\Delta y_j)^2)] \\
+\alpha_{i1,j}k_{i2,j}\Delta x_{i2}/(\alpha_{i2,j}\Delta x_{i1}k_{i1,j})(1/\Delta t+\alpha_{i2,j}((\theta_1+\theta_2)/(\Delta x_{i2})^2+(\theta_3+\theta_4)/(\Delta y_j)^2) \\
=2\alpha_{i1,j}(\theta_2T^{n+1}_{i-1,j}+(1-\theta_2)T^{n}_{i-1,j})/(\Delta x_{i1})^2+\alpha_{i1,j}(1+k_{i2,j}\Delta x_{i2}/(\Delta x_{i1}k_{i1,j})) \\
[\theta_3T^{n+1}_{i,j+1}+(1-\theta_3)T^{n}_{i,j+1}+\theta_4T^{n+1}_{i,j-1}+(1-\theta_4)T^{n}_{i,j-1}]/(\Delta y_j)^2 \\
+[1/\Delta t-\alpha_{i1,j}((2-\theta_1-\theta_2)/(\Delta x_{i1})^2+(2-\theta_3-\theta_4)/(\Delta y_j)^2)+\alpha_{i1,j}\Delta x_{i2}k_{i2,j} \\
/(\alpha_{i2,j}\Delta x_{i1}k_{i1,j})(1/\Delta t-\alpha_{i2,j}((2-\theta_1-\theta_2)/(\Delta x_{i2})^2+(2-\theta_3-\theta_4)/(\Delta y_j)^2))]T^{n}_{i,j} \\
+2\alpha_{i1,j}k_{i2,j}/(\Delta x_{i1}\Delta x_{i2}k_{i1,j})(\theta_1T^{n+1}_{i+1,j}+(1-\theta_1)T^{n}_{i+1,j}) \\
+\alpha_{i1,j}(q_{i1,j}+q_{i2,j}\Delta x_{i2}/\Delta x_{i1})/(\Delta y_jk_{i1,j})
\end{gathered} \tag{81}$$

At the corner of a heater, there is an intersection between an x-direction interface and a y-direction interface. To account for this point, an equation will be developed in the same manner as above. First, Eq. (45b), which is a y-direction interface boundary condition, is solved for the fictious temperatures $T^{n+1}_{i,j+1}{}^{\Delta}$ and $T^{n}_{i,j+1}{}^{\Delta}$ to get

$$\begin{gathered}
\theta_3T^{n+1}_{i,j+1}{}^{\Delta}+(1-\theta_3)T^{n}_{i,j+1}{}^{\Delta}=\theta_4T^{n+1}_{i,j-1}+(1-\theta_4)T^{n}_{i,j-1}+k_{i,j2}\Delta y_{j1} \\
(\theta_3T^{n+1}_{i,j+1}+(1-\theta_3)T^{n}_{i,j+1}-(\theta_4T^{n+1}_{i,j-1}+(1-\theta_4)T^{n}_{i,j-1})^{\Delta})/(k_{i,j1}\Delta y_{j2})
\end{gathered} \tag{82}$$

Equation (81) is now written for the j1 and j2 layers which are the layers that form this interface:

$$
\begin{aligned}
&T^{n+1}_{i,j}[1/\Delta t+\alpha_{i1,j1}((\theta_1+\theta_2)/(\Delta x_{i1})+(\theta_3+\theta_4)/(\Delta y_{j1})^2)+\alpha_{i1,j1}k_{i2,j1}\Delta x_{i2}\\
&/(\alpha_{i2,j1}\Delta x_{i1}k_{i1,j1})(1/\Delta t+\alpha_{i2,j1}((\theta_1+\theta_2)/(\Delta x_{i2})^2+(\theta_3+\theta_4)/(\Delta y_{j1})^2))]\\
&=2\alpha_{i1,j1}(\theta_2T^{n+1}_{i-1,j}+(1-\theta_2)T^{n}_{i-1,j})/(\Delta x_{i1})^2+\alpha_{i1,j1}(1+k_{i2,j1}\Delta x_{i2}/(\Delta x_{i1}k_{i1,j1}))\\
&[\theta_3T^{n+1}_{i,j+1}{}^{\Delta}+(1-\theta_3)T^{n}_{i,j+1}{}^{\Delta}+\theta_4T^{n+1}_{i,j-1}+(1-\theta_4)T^{n}_{i,j-1}]/(\Delta y_{j1})^2\\
&+[1/\Delta t-\alpha_{i1,j1}((2-\theta_1-\theta_2)/(\Delta x_{i1})^2+(2-\theta_3-\theta_4)/(\Delta y_{j1})^2)+\alpha_{i1,j1}\Delta x_{i2}k_{i2,j1}\\
&/(\alpha_{i2,j1}\Delta x_{i1}k_{i1,j1})(1/\Delta t-\alpha_{i2,j1}((2-\theta_1-\theta_2)/(\Delta x_{i2})^2+(2-\theta_3-\theta_4)/(\Delta y_{j1})^2))]T^{n}_{i,j}\\
&+2\alpha_{i1,j1}k_{i2,j1}/(\Delta x_{i1}\Delta x_{i2}k_{i1,j1})(\theta_1T^{n+1}_{i+1,j}+(1-\theta_1)T^{n}_{i+1,j})\\
&+\alpha_{i1,j1}(q_{i1,j1}+q_{i2,j1}\Delta x_{i2}/\Delta x_{i1})/(\Delta y_{j1}k_{i1,j1})
\end{aligned}
\tag{83a}
$$

$$
\begin{aligned}
&T^{n+1}_{i,j}[1/\Delta t+\alpha_{i1,j2}((\theta_1+\theta_2)/(\Delta x_{i1})^2+(\theta_3+\theta_4)/(\Delta y_{j2})^2+\alpha_{i1,j2}k_{i2,j2}\Delta x_{i2}\\
&/(\alpha_{i2,j2}\Delta x_{i1}k_{i1,j2})(1/\Delta t+\alpha_{i2,j2}((\theta_1+\theta_2)/(\Delta x_{i2})^2+(\theta_3+\theta_4)/(\Delta y_{j2})^2))]\\
&=2\alpha_{i1,j2}(\theta_2T^{n+1}_{i-1,j}+(1-\theta_2)T^{n}_{i-1,j})/(\Delta x_{i1})^2+\alpha_{i1,j2}(1+k_{i2,j2}\Delta x_{i2}/(\Delta x_{i1}k_{i1,j2}))\\
&\quad[\theta_3T^{n+1}_{i,j+1}+(1-\theta_3)T^{n}_{i,j+1}+\theta_4T^{n+1}_{i,j-1}{}^{\Delta}+(1-\theta_4)T^{n}_{i,j-1}{}^{\Delta}]/(\Delta y_{j2})^2\\
&+[1/\Delta t-\alpha_{i1,j2}((2-\theta_1-\theta_2)/(\Delta x_{i1})^2+(2-\theta_3-\theta_4)/(\Delta y_{j2})^2)+\alpha_{i1,j2}\Delta x_{i2}k_{i2,j2}\\
&/(\alpha_{i2,j2}\Delta x_{i1}k_{i1,j2})(1/\Delta t-\alpha_{i2,j2}((2-\theta_1-\theta_2)/(\Delta x_{i2})^2+(2-\theta_3-\theta_4)/(\Delta y_{j2})^2))]T^{n}_{i,j}\\
&+2\alpha_{i1,j2}k_{i2,j2}/(\Delta x_{i1}\Delta x_{i2}k_{i1,j2})(\theta_1T^{n+1}_{i+1,j}+(1-\theta_1)T^{n}_{i+1,j})\\
&+\alpha_{i1,j2}(q_{i1,j2}+q_{i2,j2}\Delta x_{i2}/\Delta x_{i1})/(\Delta y_{j2}k_{i1,j2})
\end{aligned}
\tag{83b}
$$

Next Eq. (82) is substituted into Eq. (83a) to eliminate $T^{n+1}_{i,j+1}{}^{\Delta}$ and $T^{n+1}_{i,j+1}{}^{\Delta}$. This results in

$$T^{n+1}_{i,j}[1/\Delta t+\alpha_{i1,j1}((\theta_1+\theta_2)/(\Delta x_{i1})^2+(\theta_3+\theta_4)/(\Delta y_{j1})^2)+\alpha_{i1,j1}k_{i2,j1}\Delta x_{i2}$$

$$/(\alpha_{i2,j1}\Delta x_{i1}k_{i1,j1})(1/\Delta t+\alpha_{i2,j1}((\theta_1+\theta_2)/(\Delta x_{i2})^2+(\theta_3+\theta_4)/(\Delta y_{j1})^2))]$$

$$=2\alpha_{i1,j1}(\theta_2T^{n+1}_{i-1,j}+(1-\theta_2)T^{n}_{i-1,j})/(\Delta x_{i1})^2+\alpha_{i1,j1}(1+k_{i2,j1}\Delta x_{i2}\ /(\Delta x_{i1}k_{i1,j1}))$$

$$[2\theta_4T^{n+1}_{i,j-1}+2(1-\theta_4)T^{n}_{i,j-1}+k_{i,j2}\Delta y_{j1}(\theta_3T^{n+1}_{i,j+1}$$

$$+(1-\theta_3)T^{n}_{i,j+1}-(\theta_4T^{n+1}_{i,j-1}+(1-\theta_4)T^{n}_{i,j-1})^{\Delta})/(k_{i,j1}\Delta y_{j2})]/(\Delta y_{j1})^2 \qquad (84)$$

$$+[1/\Delta t-\alpha_{i1,j1}((2-\theta_1-\theta_2)/(\Delta x_{i1})^2+(2-\theta_3-\theta_4)/(\Delta y_{j1})^2)+\alpha_{i1,j1}\Delta x_{i2}k_{i2,j1}$$

$$/(\alpha_{i2,j1}\Delta x_{i1}k_{i1,j1})(1/\Delta t-\alpha_{i2,j1}((2-\theta_1-\theta_2)/(\Delta x_{i2})^2+(2-\theta_3-\theta_4)/(\Delta y_{j1})^2))]T^{n}_{i,j}$$

$$+2\alpha_{i1,j1}k_{i2,j1}/(\Delta x_{i1}\Delta x_{i2}k_{i1,j1})(\theta_1T^{n+1}_{i+1,j}+(1-\theta_1)T^{n}_{i+1,j})$$

$$+\alpha_{i1,j1}(q_{i1,j1}+q_{i2,j1}\Delta x_{i2}/\Delta x_{i1})/(\Delta y_{j1}k_{i1,j1})$$

Finally, $T^{n+1}_{i,j-1}{}^{\Delta}$ and $T^{n}_{i,j-1}{}^{\Delta}$ are eliminated by substituting Eq. (83b) into Eq. (84) to obtain the equation for the intersection of a y-direction interface with an x-direction interface. This expression is

$$T^{n+1}_{i,j}\{[1/\Delta t+\alpha_{i1,j1}((\theta_1+\theta_2)/(\Delta x_{i1})^2+(\theta_3+\theta_4)/(\Delta y_{j1})^2)+\alpha_{i1,j1}k_{i2,j1}\Delta x_{i2}$$

$$/(\alpha_{i2,j1}\Delta x_{i1}k_{i1,j1})(1/\Delta t+\alpha_{i2,j1}((\theta_1+\theta_2)/(\Delta x_{i2})^2+(\theta_3+\theta_4)/(\Delta y_{j1})^2))]$$

$$+\alpha_{i1,j1}k_{i,j2}\Delta y_{j2}(1+k_{i2,j1}\Delta x_{i2}/(\Delta x_{i1}k_{i1,j1}))/(\alpha_{i1,j2}k_{i,j1}\Delta y_{j1}\ (1+k_{i2,j2}\Delta x_{i2}/(\Delta x_{i1}k_{i1,j2})))$$

$$[1/\Delta t+\alpha_{i1,j2}((\theta_1+\theta_2)/(\Delta x_{i1})^2+(\theta_3+\theta_4)/(\Delta y_{j2})^2)+\alpha_{i1,j2}k_{i2,j2}\Delta x_{i2}$$

$$/(\alpha_{i2,j2}\Delta x_{i1}k_{i1,j2})(1/\Delta t+\alpha_{i2,j2}((\theta_1+\theta_2)/(\Delta x_{i2})^2+(\theta_3+\theta_4)/(\Delta y_{j2})^2))]\}$$

$$=2\alpha_{i1,j1}(\theta_2T^{n+1}{}_{i-1,j}+(1-\theta_2)T^n{}_{i-1,j})/(\Delta x_{i1})^2+\alpha_{i1,j1}(1+k_{i2,j1}\Delta x_{i2}/(\Delta x_{i1}k_{i1,j1}))$$

$$2\alpha_{i1,j1}k_{i,j2}\Delta y_{j2}(1+k_{i2,j1}\Delta x_{i2}/(\Delta x_{i1}k_{i1,j1}))\ (\theta_2T^{n+1}{}_{i-1,j}+(1-\theta_2)T^n{}_{i-1,j})$$

$$/((\Delta x_{i1})^2k_{i,j1}\Delta y_{j1}(1+k_{i2,j2}\Delta x_{i2}/(\Delta x_{i1}k_{i1,j2})))+2\alpha_{i1,j1}[\theta_4T^{n+1}{}_{i,j-1}$$

$$+(1-\theta_4)T^n{}_{i,j-1}+k_{i,j2}\Delta y_{j1}(\theta_3T^{n+1}{}_{i,j+1}+(1-\theta_3)T^n{}_{i,j+1})/(k_{i,j1}\Delta y_{j2})]/(\Delta y_{j1})^2$$

$$+\{[1/\Delta t-\alpha_{i1,j1}((2-\theta_1-\theta_2)/(\Delta x_{i1})^2+(2-\theta_3-\theta_4)/(\Delta y_{j1})^2+\alpha_{i1,j1}\Delta x_{i2}k_{i2,j1} \qquad (85)$$

$$/(\alpha_{i2,j1}\Delta x_{i1}k_{i1,j1})(1/\Delta t-\alpha_{i2,j1}((2-\theta_1-\theta_2)/(\Delta x_{i2})^2+(2-\theta_3-\theta_4)/(\Delta y_{j1})^2))]$$

$$+\alpha_{i1,j1}k_{i,j2}\Delta y_{j2}(1+k_{i2,j1}\Delta x_{i2}/(\Delta x_{i1}k_{i1,j1}))/(\alpha_{i1,j2}k_{i,j1}\Delta y_{j1}\ (1+k_{i2,j2}\Delta x_{i2}/(\Delta x_{i1}k_{i1,j2})))$$

$$[1/\Delta t-\alpha_{i1,j2}((2-\theta_1-\theta_2)/(\Delta x_{i1})^2+(2-\theta_3-\theta_4)/(\Delta y_{j2})^2+\alpha_{i1,j2}\Delta x_{i2}k_{i2,j2}/$$

$$(\alpha_{i2,j2}\Delta x_{i1}k_{i1,j2})(1/\Delta t-\alpha_{i2,j2}((2-\theta_1-\theta_2)/(\Delta x_{i2})^2+(2-\theta_3-\theta_4)/(\Delta y_{j2})^2))]\}T^n{}_{i,j}$$

$$+[2\alpha_{i1,j1}k_{i2,j1}/(\Delta x_{i1}\Delta x_{i2}k_{i1,j1})+2k_{i2,j2}\alpha_{i1,j1}k_{i,j2}\Delta y_{j2}(1+k_{i2,j1}\Delta x_{i2}$$

$$/(\Delta x_{i1}k_{i1,j1}))/(\Delta x_{i1}\Delta x_{i2}k_{i1,j2}k_{i,j1}\Delta y_{j1}(1+k_{i2,j2}\Delta x_{i2}/(\Delta x_{i1}k_{i1,j2})))]$$

$$(\theta_1T^{n+1}{}_{i+1,j}+(1-\theta_1)T^n{}_{i+1,j})+\alpha_{i1,j1}(q_{i1,j1}+q_{i2,j1}\Delta x_{i2}/\Delta x_{i1})/(\Delta y_{j1}k_{i1,j1})$$

$$+\alpha_{i1,j1}k_{i,j2}(1+k_{i2,j1}\Delta x_{i2}/(\Delta x_{i1}k_{i1,j1}))$$

$$(q_{i1,j2}+q_{i2,j2}\Delta x_{i2}/\Delta x_{i1})/(k_{i1,j2}k_{i,j1}\Delta y_{j1}(1+k_{i2,j2}\Delta x_{i2}/(\Delta x_{i1}k_{i1,j2})))$$

This concludes the derivation of equations for the application of boundary conditions to the finite difference form of the energy equation. For any of the methods described, the collection of equations can be assembled into matrix form. These matrix equations can be solved for the unknown temperatures using a variety of solution techniques. These methods of solution will be described in the next section.

V. NUMERICAL SOLUTION METHODS

There are three categories into which the seven formulations discussed previously fall. First, there are the explicit methods in which all unknown temperatures are directly calculated from known values. Hence, there is no need for a numerical method in order to solve the matrix of unknown temperatures which arise from explicit methods since the solution can be found by direct calculation. The explicit methods which have been presented are the Simple Explicit, the Hopscotch, and the ADE (Alternating Direction Explicit) methods.

Second, there are two methods which, although implicit, can be solved without iteration. These are the ADI (Alternating Direction Implicit) and the Time-Splitting methods. These methods can be solved using tridiagonal matrix solution techniques. A tridiagonal system is one in which all three unknown temperatures are in one direction (i.e., $T^{n+1}_{i,j}$ $T^{n+1}_{i+1,j}$ and $T^{n+1}_{i-1,j}$) and can be written such that the coefficients of the temperatures lie on the three main diagonals of the coefficient matrix. The coefficient matrix can readily be put into either an upper or lower triangular form and the unknown temperatures determined by backward or forward substitution. Thus, a tridiagonal system can easily be inverted to obtain values for all unknown temperatures in the grid without the need for any iteration.

Finally, there are two methods (Crank-Niccolson and Simple Implicit) which can not be solved directly. These methods are implicit and contain five unknown temperatures in every equation. As such, an iteration (trial-and-error) method must be used to obtain a solution at each time step. There are three iteration methods which will be covered here. These are: Gauss-Seidel iteration; SIP (Strongly Implicit Procedure); and MSIP (Modified Strongly Implicit Procedure).

A. GAUSS-SEIDEL ITERATION

Although most literature sources do not consider Gauss-Seidel iteration to be as efficient as the others to be discussed, it is presented here as a historical reference.

Previous attempts to solve the deicer pad problem used a Crank-Nicolson formulation with Gauss-Seidel iteration. This approach was used because no method existed to eliminate the extra unknown of enthalpy from the ice layer equations. A technique to accomplish this will be discussed in the next section. Gauss-Seidel iteration does not need to eliminate enthalpy since the phase of each node in the ice can be updated after each iteration.

The Gauss-Seidel iteration procedure is quite simple. First, the difference equation at each point in the grid is used to solve for the temperature (or enthalpy as the case may be) at the i,j node. Approximations for the other unknown temperatures in the equation are used so that a new temperature can be found at this point. Initially, the temperature from the previous time step is used as a first-guess approximation. Updated values are used as the iteration proceeds. The iteration for that time step is completed when the maximum percent difference between any newly calculated temperature and the value from the previous iteration is less than some predefined value. A value of 10^{-5} was found to be sufficient for this problem.

This iteration process can be made faster by the use of over-relaxation. Over-relaxation is a procedure in which the value at each iteration is over-predicted in the hope that the newly calculated temperatures will approach convergence more quickly. This is accomplished by multiplying the finite-differenced temperature equation by a parameter ω, which is called the over-relaxation parameter, and then subtracting ω-1 times the previous temperature at this node. Using the Crank-Nicolson formulation for a point in the interior of the grid as an example, this process can be expressed by the following equation:

$$\begin{aligned}T^{n+1}_{i,j}=\omega[&\alpha_{i,j}(T^{n+1}_{i+1,j}+T^{n}_{i+1,j}+T^{n+1}_{i-1,j}+T^{n}_{i-1,j})/2(\Delta x_i)^2\\&+\alpha_{i,j}(T^{n+1}_{i,j+1}+T^{n}_{i,j+1}+T^{n+1}_{i,j-1}+T^{n}_{i,j-1})/2(\Delta y_j)^2\qquad(86)\\&+(1/\Delta t-\alpha_{i,j}/(\Delta x_i)^2-\alpha_{i,j}/(\Delta y_j)^2)T^{n}_{i,j}]/[1/\Delta t+\alpha_{i,j}/(\Delta x_i)^2+\alpha_{i,j}/(\Delta y_j)^2]+(1-\omega)T^{n}_{i,j}\end{aligned}$$

Notice that in the above equation if ω is equal to 1, the original formulation is obtained. This over-relaxation parameter must also be between 1 and 2 for this procedure to work. A value of 1.7 was found to be the most efficient for this problem.

B. SIP AND MSIP METHODS

The remaining solution methods, SIP and MSIP, are quite similar to each other. In fact, MSIP is a modification of SIP. These procedures solve the matrix of unknown temperatures by factoring the equations into two matrices, each of which can be easily inverted. First, the equation for temperature at each point is written in the form

$$[A]\, T^{n+1}_{i+1,j} + [B]\, T^{n+1}_{i-1,j} + [C]\, T^{n+1}_{i,j+1} + [D]\, T^{n+1}_{i,j-1} + [E]\, T^{n+1}_{i,j} = [F] \tag{87}$$

where [A], [B], [C], [D], [E], and [F] are known values. This can also be written in the form $[M]\,[T^{n+1}] = [F]$ where [M] is the matrix of coefficients of the unknown temperatures, $[T^{n+1}]$ is the column matrix of unknown temperatures, and [F] is a column matrix of known coefficients and temperatures. To this equation is added some small matrix [N] such that $[P]\,[T^{n+1}] = [F] + [N]\,[T^{n+1}]$, where the matrix [P] is equal to [M+N]. This matrix can now be divided into a lower and upper diagonal matrix, i.e., $[P] = [L][U]$. The upper and lower diagonal matrices can each be inverted easily to obtain a solution for $[T^{n+1}]$. The matrix [N] which was added to [M] is defined in such a way as to make this practical. The only difference between SIP and MSIP is how the matrix [N] is found. In the equation $[P]\,[T^{n+1}] = [F] + [N]\,[T^{n+1}]$, there are unknown temperatures on both sides of the equation; hence, a small amount of iteration is needed to find the final solution for a single time step. The solution algorithm for this is given below.

To begin,

$$[P]\,[T^{n+1}] = [F] + [N]\,[T^{n+1}] \tag{88}$$

Let

$$[N]\,[T^{n+1}] = [N]\,[T^{n}] \tag{89}$$

Then

$$[P]\,[T^{n+1}] = [F] + [N]\,[T^{n}] \tag{90}$$

Now add and subtract [M] $[T^n]$

$$[P]\,[T^{n+1}] = [F] + [M+N]\,[T^{n+1}]-[M]\,[T^n] \tag{91}$$

Define

$$[\delta^{n+1}]= [T^{n+1}]-[T^n] \tag{92}$$

Then

$$[P]\,[\delta^{n+1}] = [F] - [M]\,[T^n] \tag{93}$$

Or,

$$[L][U]\,[\delta^{n+1}] = [F] - [M]\,[T^n] \tag{94}$$

Define

$$[V^{n+1}]=[U]\,[\delta^{n+1}] \tag{95}$$

Then

$$[L]\,[V^{n+1}] = [F] - [M]\,[T^n]$$

and $$[V^{n+1}]=[U]\,[\delta^{n+1}] \tag{96}$$

The solution proceeds as follows: the matrix [L] is inverted to find the value of $[V^{n+1}]$ at each point; then the matrix [U] is inverted to find $[\delta^{n+1}]$ at each point; third, the value of $[T^{n+1}]$ is found from $[\delta^{n+1}]$; finally, since the assumption $[N]\,[T^{n+1}] = [N]\,[T^n]$ was made, the values of $[T^n]$ in this equation are changed to the values of $[T^{n+1}]$ which have just been calculated and the process repeated until a conveged solution is obtained. Convergence criteria of 0.001 for MSIP and 0.0001 for SIP were used. Hopefully, since the matrix [N] is small, few iterations will be necessary.

SIP defines the matrix multiplication [N] [T^{n+1}] as

$$[C]\,\{T^{n+1}_{i+1,j-1}-\alpha(T^{n+1}_{i+1,j}+T^{n+1}_{i,j-1}-T^{n+1}_{i,j})\}+[G]\,\{T^{n+1}_{i-1,j+1}-\alpha(T^{n+1}_{i,j+1}+T^{n+1}_{i-1,j}-T^{n+1}_{i,j})\} \tag{97}$$

where α is a parameter between 0 and 1, and is used to dampen the effect of [N] on [M]. The matrices [C] and [G] are found from the equation [L][U]-[M]=[N]. The elements of matrices [L] and [U] can be found from this equation as well, in terms of the elements of [M]. Similarly, MSIP defines [N] [T^{n+1}] as

$$[C]\,\{T^{n+1}_{i+2,j-1}-\alpha(2T^{n+1}_{i+1,j}+T^{n+1}_{i,j-1}-2T^{n+1}_{i,j}\}+[G]\,\{T^{n+1}_{i-2,j+1}-\alpha(2T^{n+1}_{i-1,j}+T^{n+1}_{i,j+1}-2T^{n+1}_{i,j})\} \tag{98}$$

The matrices [C] and [G] are different for MSIP but are found in the same manner. However, the parameter α is the same for both SIP and MSIP. A derivation of [C], [G], [L], and [U] for SIP and MSIP can be found in Anderson, et al. (17). As stated earlier, SIP, MSIP, and tridiagonal systems all require that enthalpy be removed from the ice layer equations before they can proceed. The next section will deal with a technique for efficiently finding the correct phase for each node in the ice. This technique is called the Method of Assumed States.

C. METHOD OF ASSUMED STATES

The Method of Assumed States allows the elimination of enthalpy from the ice layer equations by assuming a phase for each node in the ice and using a known enthalpy-temperature relationship to remove enthalpy in favor of temperature. Without knowing the phase distribution beforehand, enthalpy can not be eliminated. After this is accomplished, a solution for the temperature is found using one of the techniques described above. This solution is then checked to see if the assumed phase is correct at each nodal position. If at any node this assumption is wrong, the phase at this point is updated and a new solution is found. It has been found that very few iterations are needed to find the correct phase distribution with this procedure, since there can be only three phases (solid, melt, and liquid).

There are two rules which govern the use of this procedure. Both are easily implemented and reflect the physics of the phase change problem.

Rule One: In any single iteration, a node can not go directly from solid to liquid or from liquid to solid without passing through the melt region. This is understandable since it is very unlikely during a time step for a node to gain enough energy to bridge the large transition region for ice.

Rule Two: If the nodes which neighbor a node in the transition phase are both solid or both liquid, then the node in the melt phase is incorrect. Physically, this also makes sense since a single node can not be an energy sink or source by itself, i.e., it can not melt without one of the surrounding nodes melting as well.

The enthalpy-temperature relationship given by Eq. (4) must now be modified for this procedure to be used. A one-to-one relationship between enthalpy and temperature is needed which is not present in the melt phase as presently formulated. Multiple values of enthalpy are present at the melt temperature. This one-to-one relationship is accomplished by assuming that the ice melts over a small range of temperatures rather than at a single temperature. As long as this range is reasonably small (10^{-4} °F was used throughout this study), the accuracy of the solution is not significantly altered. The expressions below indicate the modified relationship between enthalpy and temperature:

$$T = \begin{cases} H/\rho_s C_{ps} & H < H_{sm} \\ \Delta T_m (H - H_{sm}) / (H_{lm} - H_{sm}) + T_m & H_{sm} < H < H_{lm} \\ (H - H_{lm})/\rho_L C_{pL} + T_m & H > H_{lm} \end{cases} \tag{99}$$

with

$$H_{sm} = \rho_s C_{ps} T_m \tag{5a}$$

$$H_{lm} = \rho_L (C_{ps} T_m + L_f) \tag{5b}$$

VI. DISCUSSION OF RESULTS

In the previous sections, numerous numerical methods have been presented, all of which are capable of solving the equations describing the transient two-dimensional heat transfer in an electrothermal deicer pad. These methods will now be evaluated to determine which method is the most efficient for this problem. First, the parameters for an accurate solution must be determined for each of these methods. This is needed prior to a comparison of computer times since it does not matter how fast a particular method is unless the solution obtained is reasonably accurate.

Following this analysis, the method which has been shown to give an accurate solution in the least amount of computational time will be run to show the effect of different parameters on the solution. A significant objective of this section will be to determine when a two-dimensional analysis is necessary rather than a one-dimensional model. Since a one-dimensional program will obtain results using less computational time than a two-dimensional program, this analysis will also be useful in decreasing the computational time needed for the problem.

Finally, a comparison will be made with experimental data obtained by Leffel (7). This case illustrates a number of the improvements this work has made over previous computer models. Six heaters are modeled at one time, whereas only one heater could be modeled before. Each of these heaters has a different firing sequence as well as different power densities. A variable ice shape is also modeled in this case.

A. ACCURACY OF METHODS

The accuracy of each finite difference method and each solution technique must first be determined to find the appropriate time step and convergence criteria for each method. An accurate solution will be defined as a solution which, at every node, is within 1°F of the solution obtained for a predetermined standard. Since previous work has been performed with the Crank-Nicolson formulation and Gauss-Seidel iteration, this will be used as the standard. For this method, a time step of 0.1 sec and a convergence criteria of 10^{-5} has been found to yield excellent results for analytical cases (6).

Two cases were used to determine the necessary parameters for accuracy. Each case simulates a standard deicer with a quarter-inch of ice on top of the abrasion shield. A standard deicer, which is shown in Fig. (2), contains six layers including the ice layer. The heater, which is in iayer 3, is 0.25 in. wide and has a heater gap of 0.25 inches. Physical properties for the standard deicer pad are given in Table 2. Each

TABLE 2

Physical Properties of a Standard Deicer

Layer	Thickness(in)	Thermal Conductivity (BTU/hr-ft^2-^{o}F)	Thermal Diffusivity (ft^2/hr)
Aluminum D-Spar	0.087	66.50	1.6500
Epoxy/Glass Insulation	0.050	0.22	0.0087
Heater Element	0.004	7.60	0.1380
Epoxy/Glass Insulation	0.010	0.22	0.0087
Stainless Steel Abrasion Shield	0.012	8.70	0.1500
Ice	0.250*	1.29	0.0446

Heater Power Density = 30. W/in^2*
Outer Convection Coefficient = 150. BTU/hr-ft^2-^{o}F*
Inner Convection Coefficient = 1. BTU/hr-ft^2-^{o}F*
Ambient Temperature = 10. ^{o}F*
Heater Length = 0.25 in.*
Heater Gap Length = 0.25 in.*

* unless specified in description of run

simulation was run for 20 seconds of real time using a heater power density of 30 W/in^2. The first case contains one heater and uses a grid with 12 nodes in the x-direction and 49 nodes in the y-direction. The second case has three heaters with 34 nodes in the x-direction and 49 nodes in the y-direction.

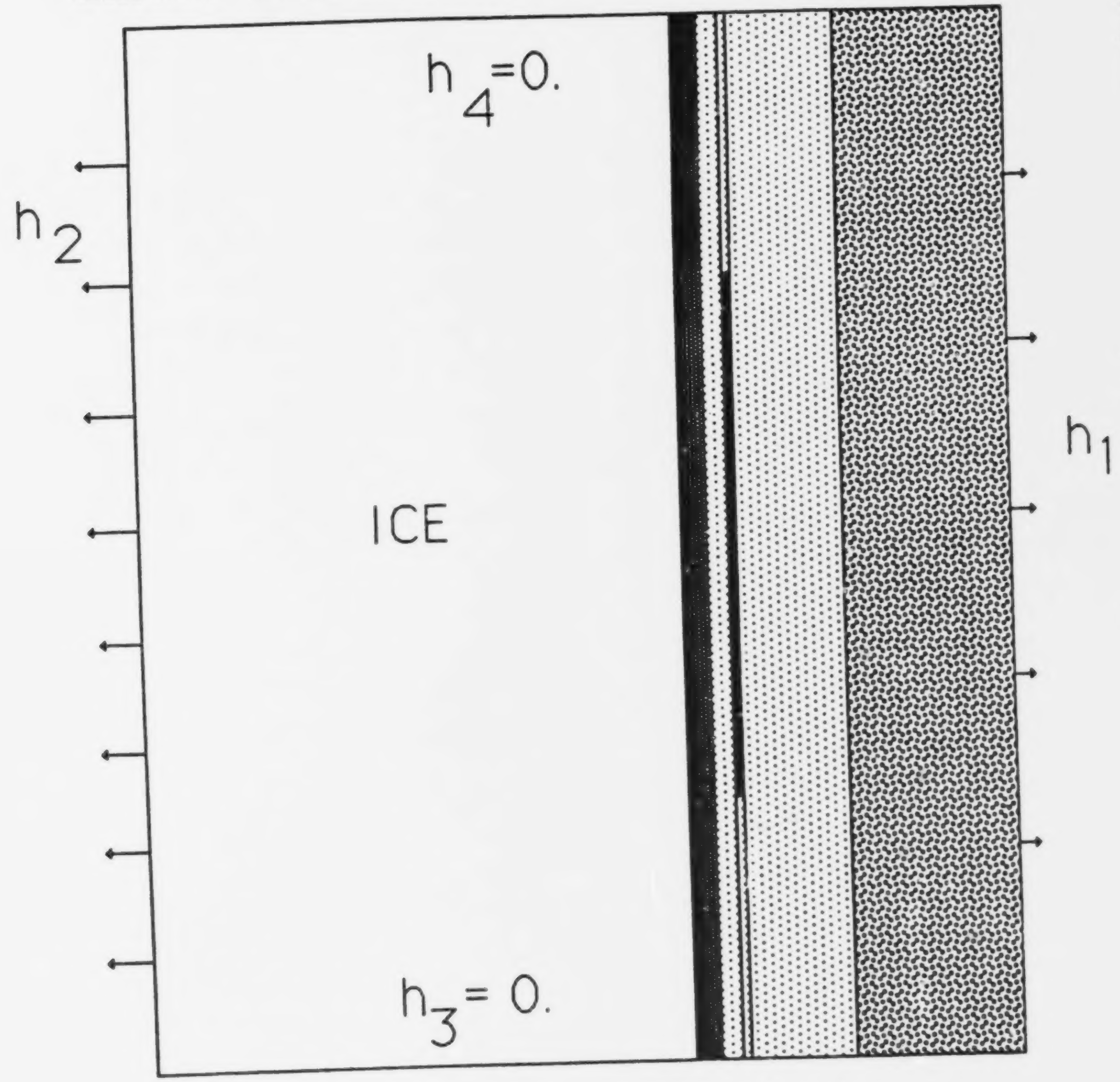

$h_4=0.$
h_2
ICE
h_1
$h_3= 0.$
▨ Substrate
▨ Insulation
■ Abrasion Shield
■ Heater Element

Figure 2

Standard Deicer Pad

For each of the methods described in the previous sections, a time step was determined by first using a large time step (1 sec.), and then subsequently decreasing this time step until the results agreed with the Crank-Nicolson solution. If the method required a convergence criteria, this was parametered after a time step was found for the method. For those methods which divide the time step into two one-half time steps, the time step given is for the half-time step, not the full time step. An even number of time steps must therefore be given for these methods. Table 3 gives the necessary time step and convergence criteria (if applicable) for each method.

Table 3

Criteria for Accurate Numerical Results

Numerical Method	Solution Technique	Time Step	Convergence Criteria
ADE	Direct	.001 sec.	N.A.(not applicable)
ADI	TDMA	.500 sec.	N.A.
Crank-Nicolson	Gauss-Seidel	.500 sec.	10^{-5}
Crank-Nicolson	MSIP	.500 sec.	10^{-3}
Crank-Nicolson	SIP	.500 sec.	10^{-4}
Hopscotch	Direct	.001 sec.	N.A.
Simple Explicit	Direct	10^{-5} sec.	N.A.
Simple Implicit	Gauss-Seidel	.500 sec.	10^{-5}
Time-Splitting	TDMA	.001 sec.	N.A.

The previous values represent the maximum time step which will yield an accurate result and the maximum convergence criteria which can be used for this time step. For the Simple Explicit method, the small time step is a reflection of the stability requirements for the two cases run. It is unknown why Time-Splitting, ADE, and Hopscotch would not give accurate results unless run with a small time step relative to the other methods, even though these are supposed to be unconditionally stable schemes. The computer time necessary to achieve an accurate solution at these values can now be compared to determine the most computationally efficient method.

B. PROGRAM EFFICIENCY

The times given in Tables 4 and 5 represent the amount of computer time expended by a NAS9080 computer for the two cases described previously. The ratio of computer times is also listed, again using Crank-Nicolson with Gauss-Seidel iteration and no over-relaxation as a basis.

Table 4

First Computer Case

Numerical Method	Solution Technique	Computer Time	Ratio
ADI	TDMA	15 sec.	0.055
Crank-Nicolson	MSIP	31 sec.	0.113
Crank-Nicolson	SIP	90 sec.	0.327
Crank-Nicolson	Gauss-Seidel with Over-Relaxation	87 sec.	0.316
Simple Implicit	Gauss-Seidel with Over-Relaxation	122 sec.	0.444
Crank-Nicolson	Gauss-Seidel with no Over-Relaxation	275 sec.	1.0
Time-Splitting	TDMA	3600 + sec.	N.A.
ADE	Direct	3600 + sec.	N.A.
Hopscotch	Direct	3600 + sec.	N.A.
Simple Explicit	Direct	3600 + sec.	N.A.

As may be seen, the ADI (Alternating Direction Implicit) method is clearly the most efficient procedure for this problem. It is approximately twice as fast as its nearest competitor, MSIP, and is up to 25 times faster than the original formulation. ADI is faster than the other methods for two main reasons. First, ADI is a direct solution method and therefore does not have to iterate to find a solution. As stated previously, direct methods can be used for the phase change problem presented because of the Method of Assumed States. Second, the time step necessary to obtain an accurate solution with this method is larger than that used for most of the other methods presented. Thus, ADI will be used in the following discussion with the knowledge that any of the methods studied would yield similar results if run using the limitations given.

Table 5

Second Computer Case

Numerical Method	Solution Technique	Computer Time	Ratio
ADI	TDMA	36 sec.	.041
Crank-Nicolson	MSIP	60 sec.	.068
Crank-Nicolson	SIP	150 sec.	.170
Crank-Nicolson	Gauss-Seidel with Over-Relaxation	222 sec	.251
Simple Implicit	Gauss-Seidel with Over-Relaxation	315 sec.	.356
Crank-Nicolson	Gauss-Seidel with no Over-Relaxation	885 sec.	1.0
Time-Splitting	TDMA	3600 + sec.	N.A.
ADE	Direct	3600 + sec.	N.A.
Hopscotch	Direct	3600 + sec.	N.A.
Simple Explicit	Direct	3600 + sec.	N.A.

C. PARAMETER VARIATION

In this section, a parametric study of some of the key variables of the problem will be performed in order to determine their effect on the solution. Since a multitude of variables could be studied, this discussion will center on those variables which are responsible for making the problem two-dimensional. These variables are chosen so that a determination can be made as to when the two-dimensional nature of the problem disallows use of a one-dimensional program. If a one-dimensional model is adequate for a particular problem, the use of a one-dimensional code such as those developed by Marano (3) or Roelke (5) will decrease the amount of computer time needed to solve the problem. Using the current approach, i.e., the ADI method, the difference in computational times between one-dimensional and two-dimensional models has been greatly reduced, but it is still significant for a reasonably complex problem.

There are four variables which determine the two-dimensional nature of the problem. First and foremost, the size of the heater gap and the physical properties of the gap play a large role in determining if the problem is two-dimensional. Second, the grid spacing may affect the solution if an inadequate number of nodes is present. A determination of the number of nodes necessary for solution is also necessary for the purpose of determining an accurate solution. Third, a variable ice shape can significantly alter the temperature profiles, especially if there are regions on the blade which have no ice. Finally, a variable outer ambient heat transfer coefficient can have an impact on the dimensionality of the problem and therefore needs to be investigated.

1. Gap Size and Material

The size of the heater gap is the primary determinant of the extent of the two-dimensional effect in a deicer pad. An analysis of heater gap thickness can therefore determine when a two-dimensional model is needed or when the faster one-dimensional models developed by Marano (3) or Roelke (5) are adequate. The material used in this heater gap is also critical since this can drastically alter the temperature of the heater. For simplicity, Chao (6) used the same material in the gap as was present in the heater. The current analysis will show when this is not a good approximation.

Again, a standard deicer is simulated with a power density of 30 W/in^2 for 20 seconds of real time. The width of the heater is kept at 0.25 inches and the gap is varied from 0.05 to 0.5 inches in increments of 0.05 inches. Figure (3) presents the temperature vs time at the centerline of the heater for the case where the physical properties of the gap are the same as the surrounding insulation. This is a more realistic case than using the properties of the heater in this section which Chao's code considers. The same cases are run using the properties of the heater in the gap, and are shown in Fig. (4). As can be seen in these plots, significant deviation from the one dimensional result (zero gap thickness) is apparent even at small gap thicknesses. This deviation is larger for the second cases which use the same conductivity in the gap than the cases which use a lower conductivity. These two cases are compared directly in Fig. (5), which plots heater temperature after 20 seconds of simluated time versus gap thickness for both sets of cases considered.

A similar set of plots are made which plot gap centerline temperature instead of heater centerline temperature for the same two sets of cases described previously. These plots (Figs. (6-8)) show an even greater dependence on the gap size, as well as a greater dependence on the properties used in the gap. The reason that gap temperatures are significant is that it may be necessary for the entire bottom layer of the ice to melt for the adhesion of the ice to the metal abrasion shield to be broken. Once this adhesion is broken, dynamic forces will remove the ice from the surface. If the gap temperatures are severely affected by the size of the gap, it will take longer to remove the ice from the surface because the temperatures at the ice-abrasion shield interface will be likewise affected.

A two-dimensional analysis is needed to accurately model the problem when the ratio of the heater length to the gap length is less than 2.5. For heater-to-gap ratios larger than this, it is felt that the decrease in heater and gap temperatures is not large enough to seriously affect the rate of ice removal. For ratios smaller than this, the combination of increasing gap size, and thus, increased ice adhesion to the surface and decreased temperatures at the surface make the use of a two-dimensional model

Figure 3

Variation in Heater Temperature with Gap Size (k gap ≠ k heater)

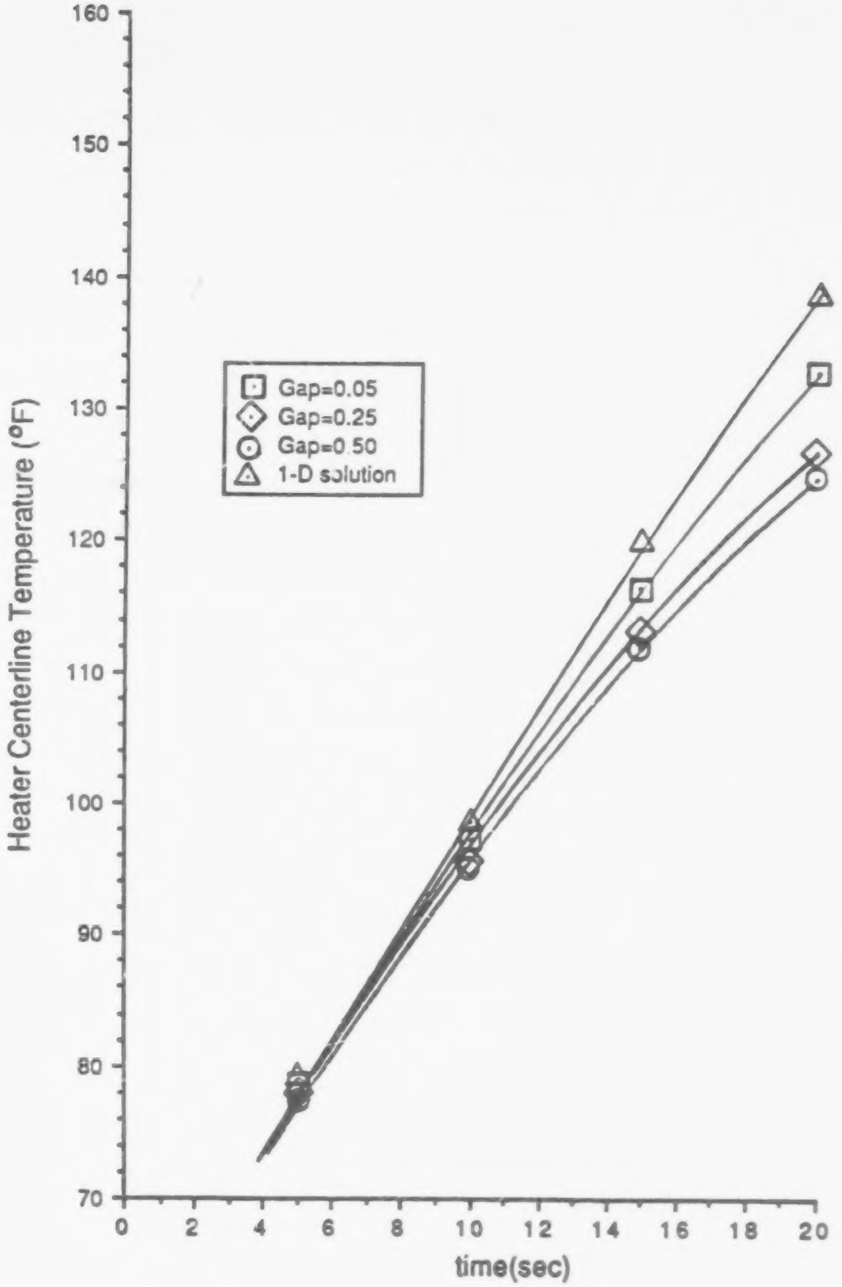


Figure 4

Variation in Heater Temperature with Gap Size (k gap = k heater)

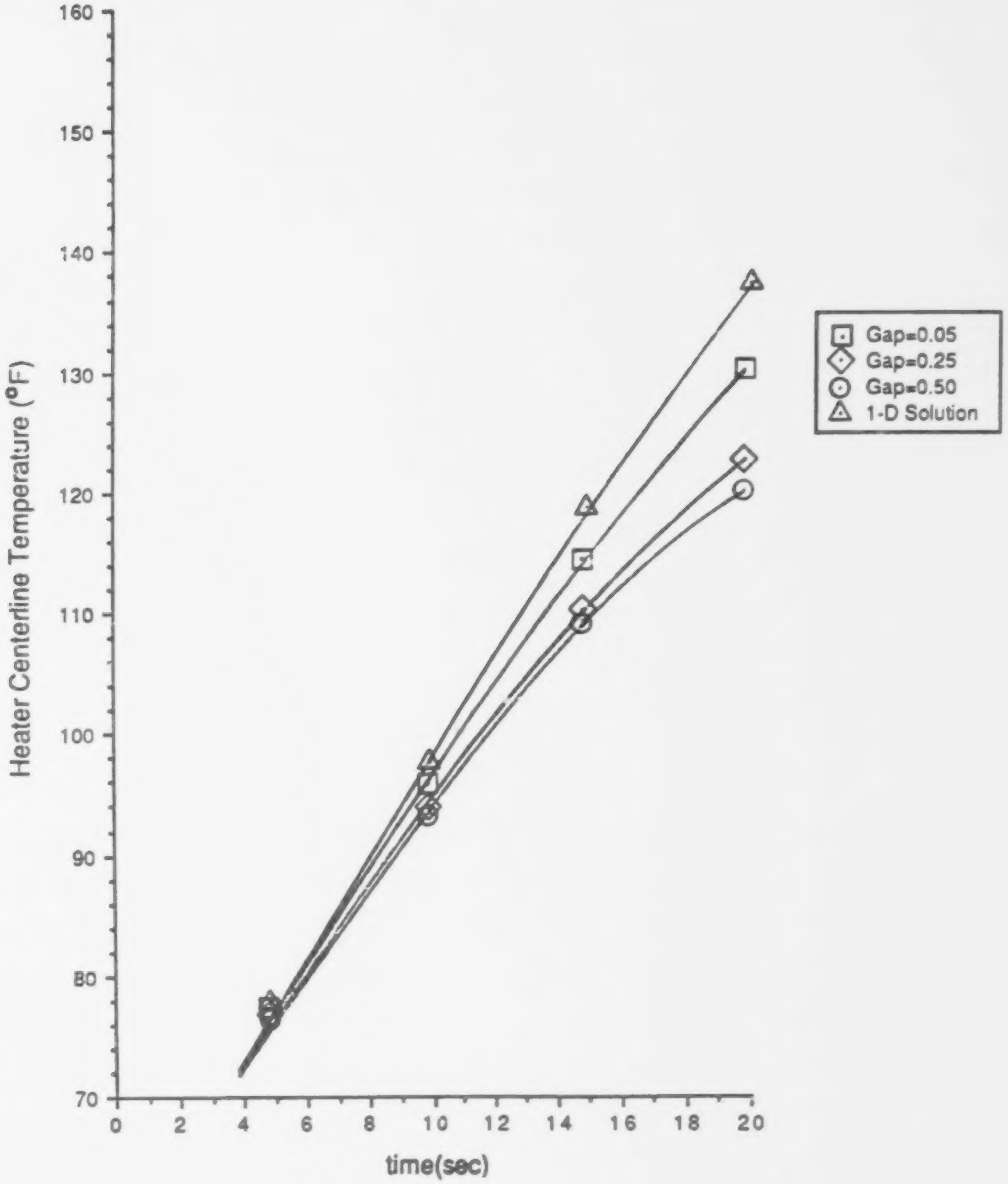
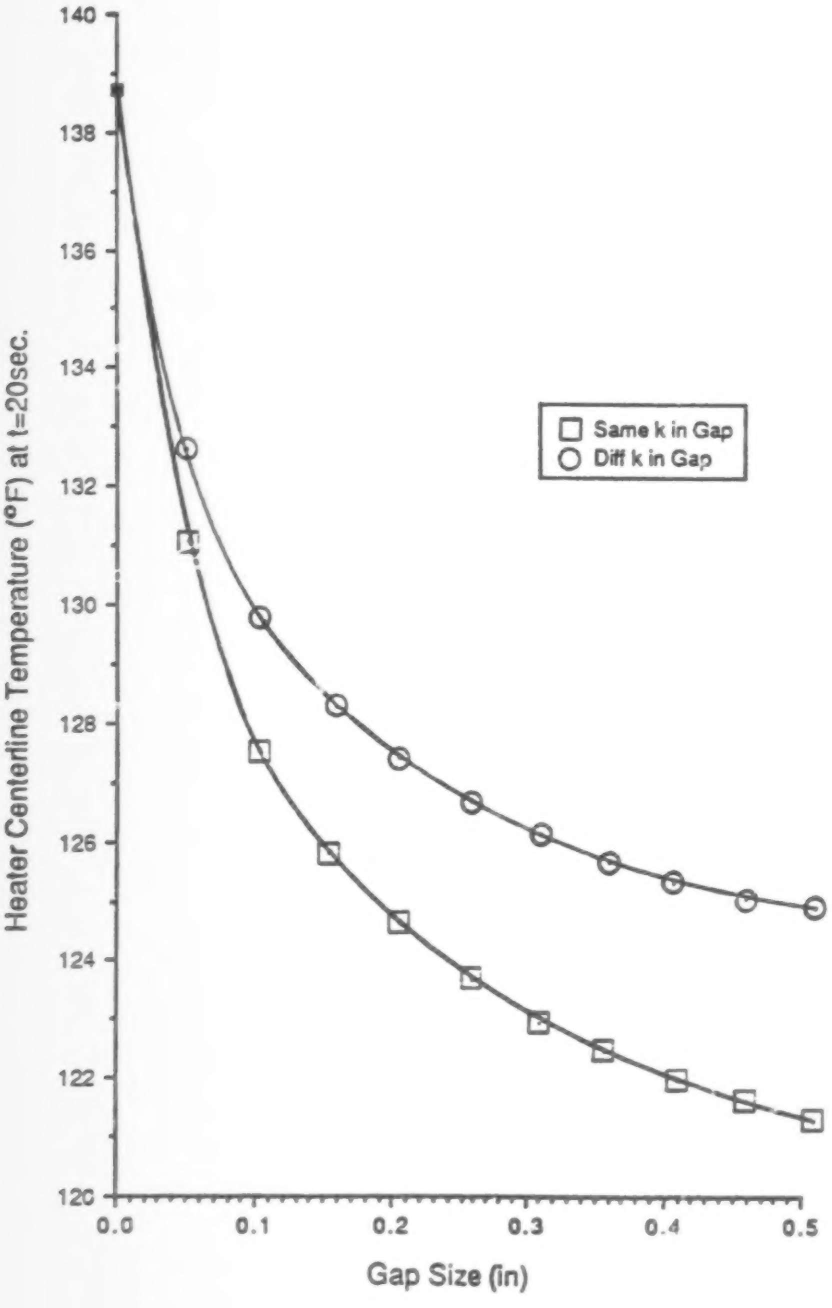


Figure 5

Effect of Gap Conductivity on Heater Temperature



7

Figure 6

Variation in Gap Temperature with Gap Size (k gap ≠ k heater)

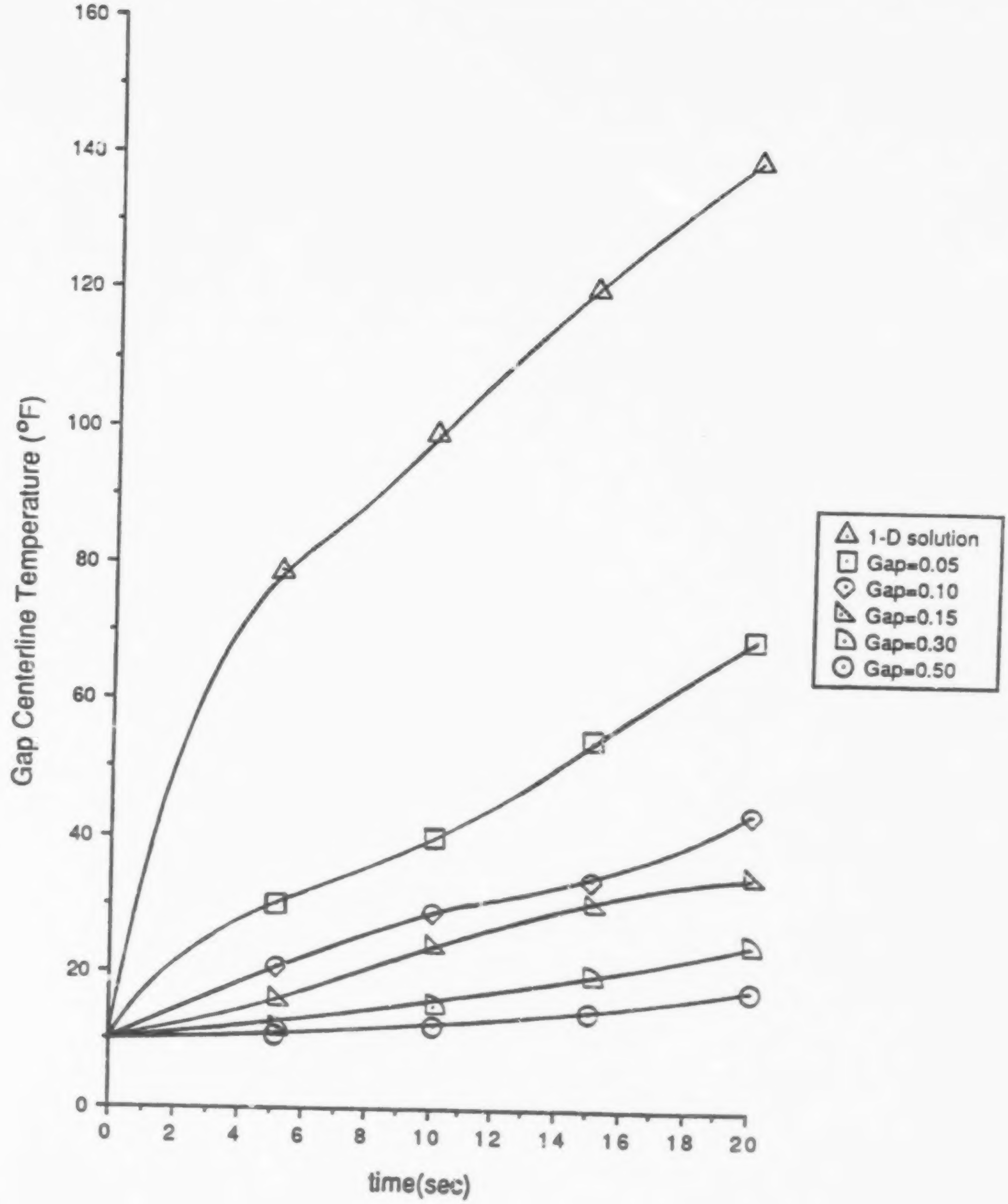


Figure 7

Variation in Gap Temperature with Gap Size (k gap = k heater)

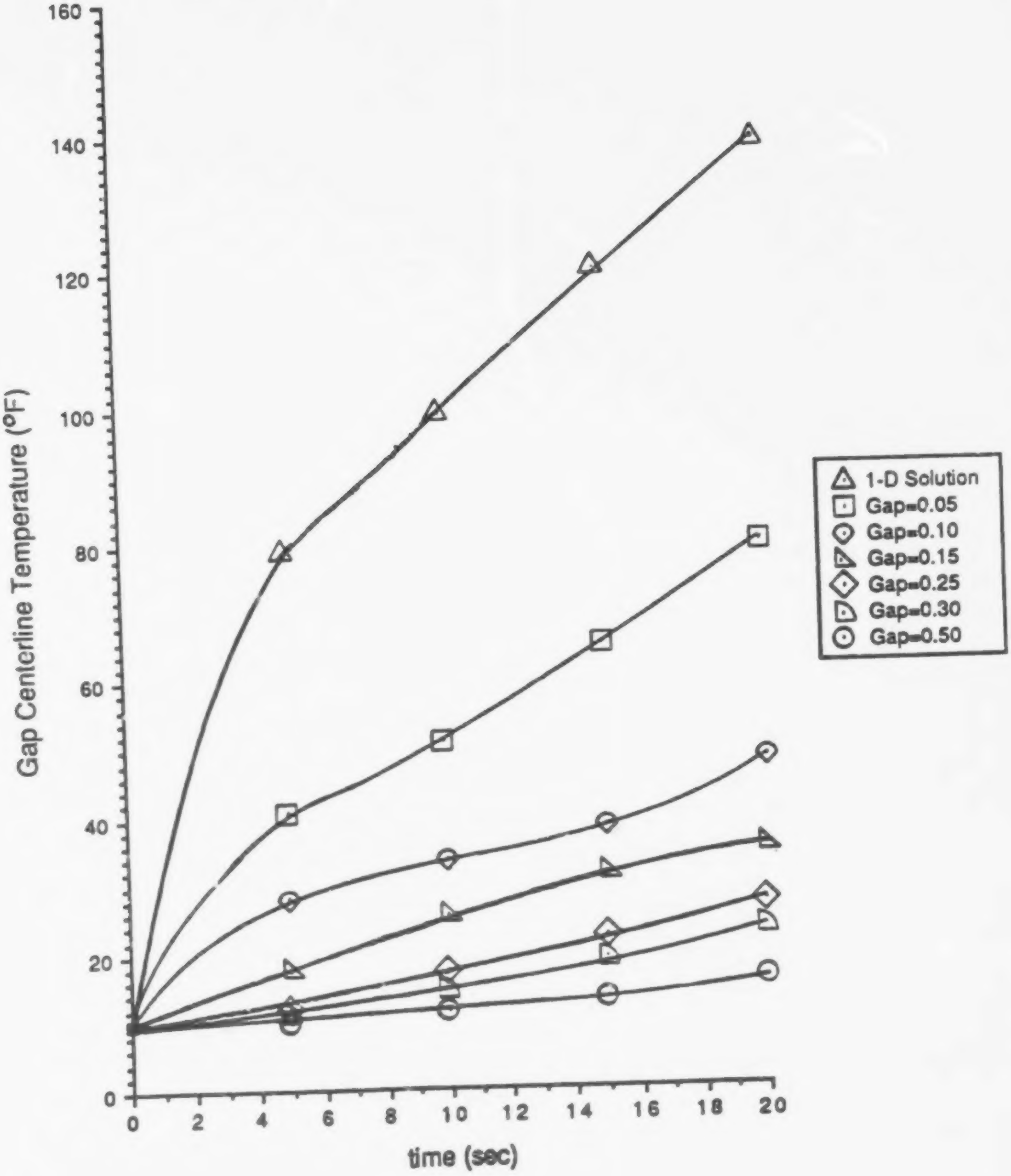
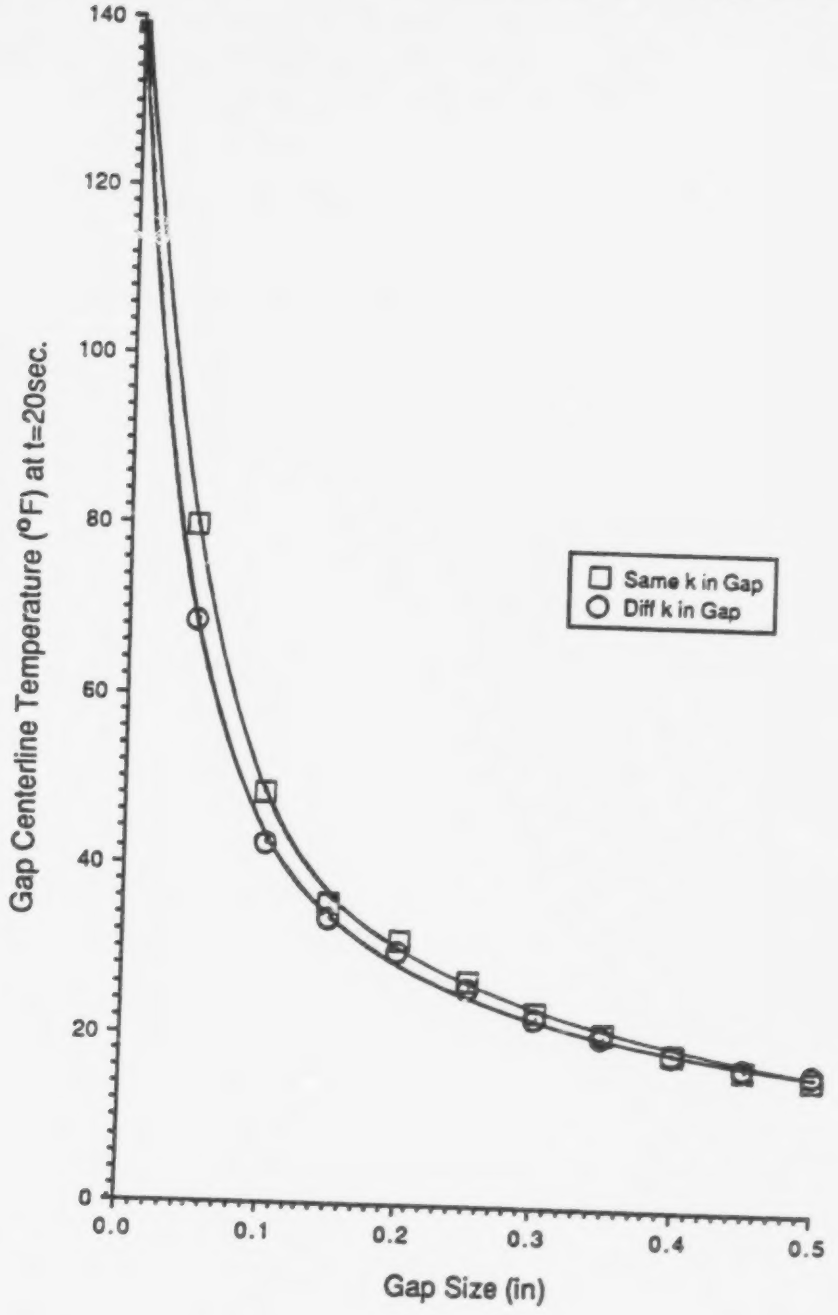


Figure 8

Effect of Gap Conductivity on Gap Temperature



necessary. Furthermore, the gap properties should always be modeled using the physical properties of the surrounding insulation, as the use of different properties in the gap has been shown to be a significant parameter in determining heater temperatures.

2. Grid Spacing

The number of nodes in both the heater and the gap were varied to study what effect, if any, this had on the results. The same cases that were used for determining the effect of gap size were run using from three to twenty nodes in the heater, and from one to twenty nodes in the gap. For all of the cases studied, six nodes were found to be necessary in the heater in the x-direction, regardless of its size. Four nodes were necessary in the gap for large gap sizes (heater length/gap length < 1), but as few as two nodes could be used in the gap for small gap sizes (heater length/gap length > 5).

It was found that the grid spacing in the heater in the y-direction should be at least an order of magnitude smaller than it is in the x-direction. This is especially true for large heater-to-gap ratios. This is understandable since the majority of the heat flow is in the y-direction and the largest temperature gradients will be found in the y-direction. A finer grid in regions which have high temperature gradients is necessary to ensure an accurate solution.

3. Ice Thickness

The size and shape of the ice layer can have a significant effect on the temperature profiles in the deicer pad, especially on the temperature at the interface between the pad and the ice. The same standard deicer case as described previously was run using various ice thicknesses and various outer heat transfer coefficients in order to determine at what point, if any, these parameters cease to have an impact on the results. This becomes especially important in modeling cases which have variable ice shapes and variable heat transfer coefficients. It is important to determine how precise each of these parameters needs to be measured. These cases, however, do not contain a gap in the heater so that the effect of these parameters on the solution could be distinguished from the effect of the gap size on the solution.

Figure (9) shows a plot of temperature vs. time at the ice-abrasion shield interface for four different ice thicknesses and for four different outer heat transfer coefficients, for a total of 16 cases. The four ice thicknesses used were 0.03125, 0.0625, 0.125, and 0.25 inches. The four heat transfer coefficients used were 40, 80, 120, and 160 BTU/hr-ft^2-^{o}F. As can be seen, the ice-interface temperature decreases slightly when the ice thickness increases for a constant heat transfer coefficient. For the same ice thickness, an increase in the heat transfer coefficient decreases the interface temperature. For ice thicknesses greater than 0.125 in., there is no effect of heat transfer coefficient on the interface temperature as a function of time. In essence, Fig. (9) is a one-dimensional result.

Variable ice thickness cases and variable heat transfer coefficient cases were also run with the standard deicer described previously. Figure (10) shows the variation in the ice-abrasion shield interface temperature after 20 sec. as a function of ice thickness for heat transfer coefficients of 40, 80, 120, and 160 BTU/hr -ft^2-^{o}F, where these heat transfer coefficients are constant for that particular run, but ice thickness is a function of x-position. For an ice thickness of 0.05 in., the ice interface temperature after 20 sec. decreases from 52.7 ^{o}F to 43.3 ^{o}F as the heat transfer coefficient changes from 40 to 160 BTU/hr -ft^2-^{o}F. Approximately the same variation occurs at other ice thicknesses. The two-dimensionality of the problem is evident from the difference in temperature magnitude shown by the curves.

Figure (11) shows a similar plot, but plots the ice-abrasion shield interface temperature after 20 sec. as a function of heat transfer coefficient for constant ice thicknesses of 0.0625, 0.125, and 0.25 inches. In this plot, the outer heat transfer coefficient is a function of the x-position. Again, the two-dimensionality of the problem is shown by the different temperature levels. For a heat transfer coefficient of 40 BTU/hr -ft^2-^{o}F, the ice interface temperature after 20 sec. drops from 52.7 ^{o}F to 47 ^{o}F as the ice thickness increases from 0.0625 in. to 0.25 inches. At a constant ice thickness, very slight interfacial temperature variation is seen as the heat transfer coefficient increases for ice thicknesses greater than 0.125 inches.

Figure 9

Effect of Ice Thickness and Heat Transfer Coefficient

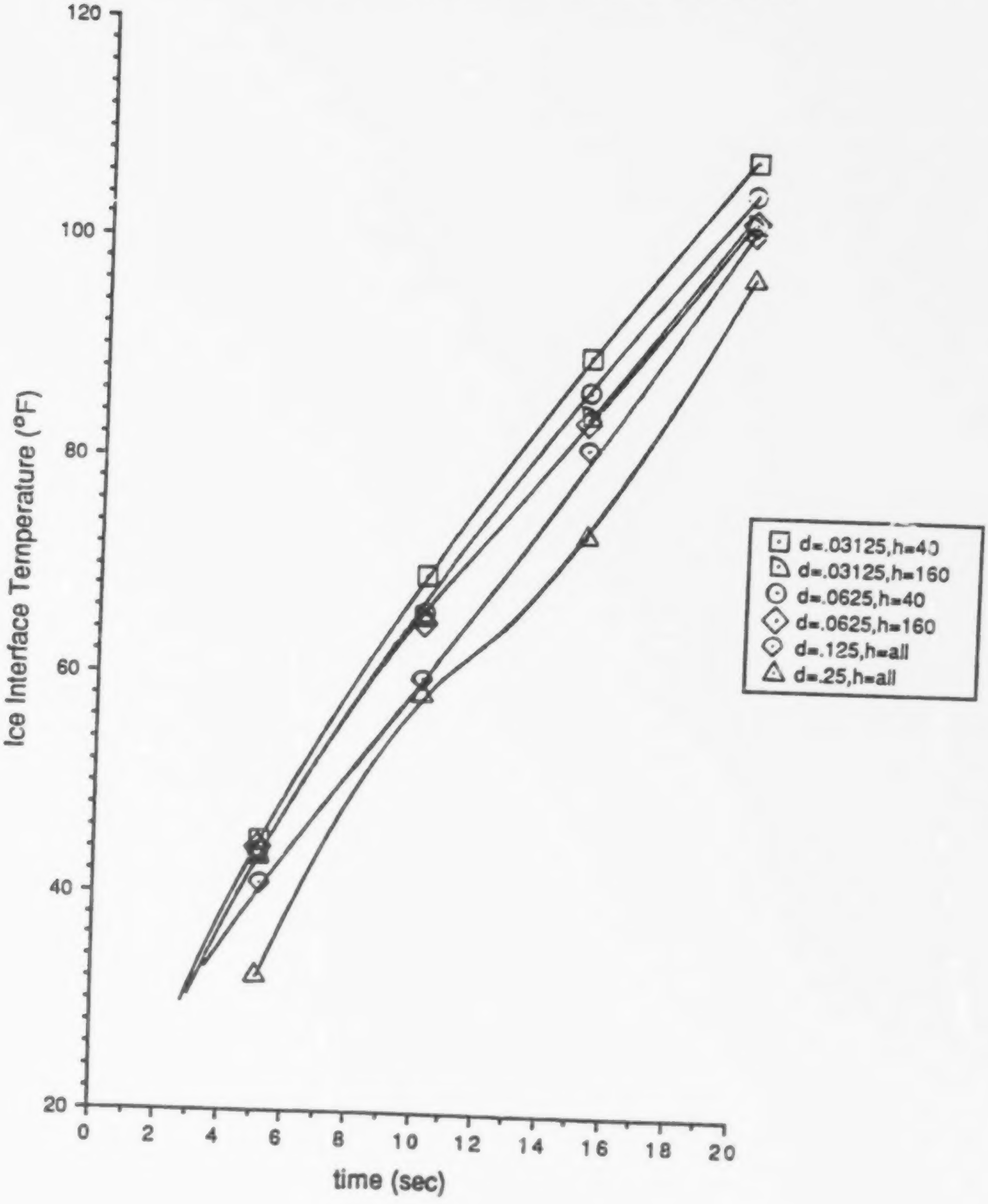


Figure 10

Effect of Variable Ice Thickness on Ice-Abrasion Shield Temperature

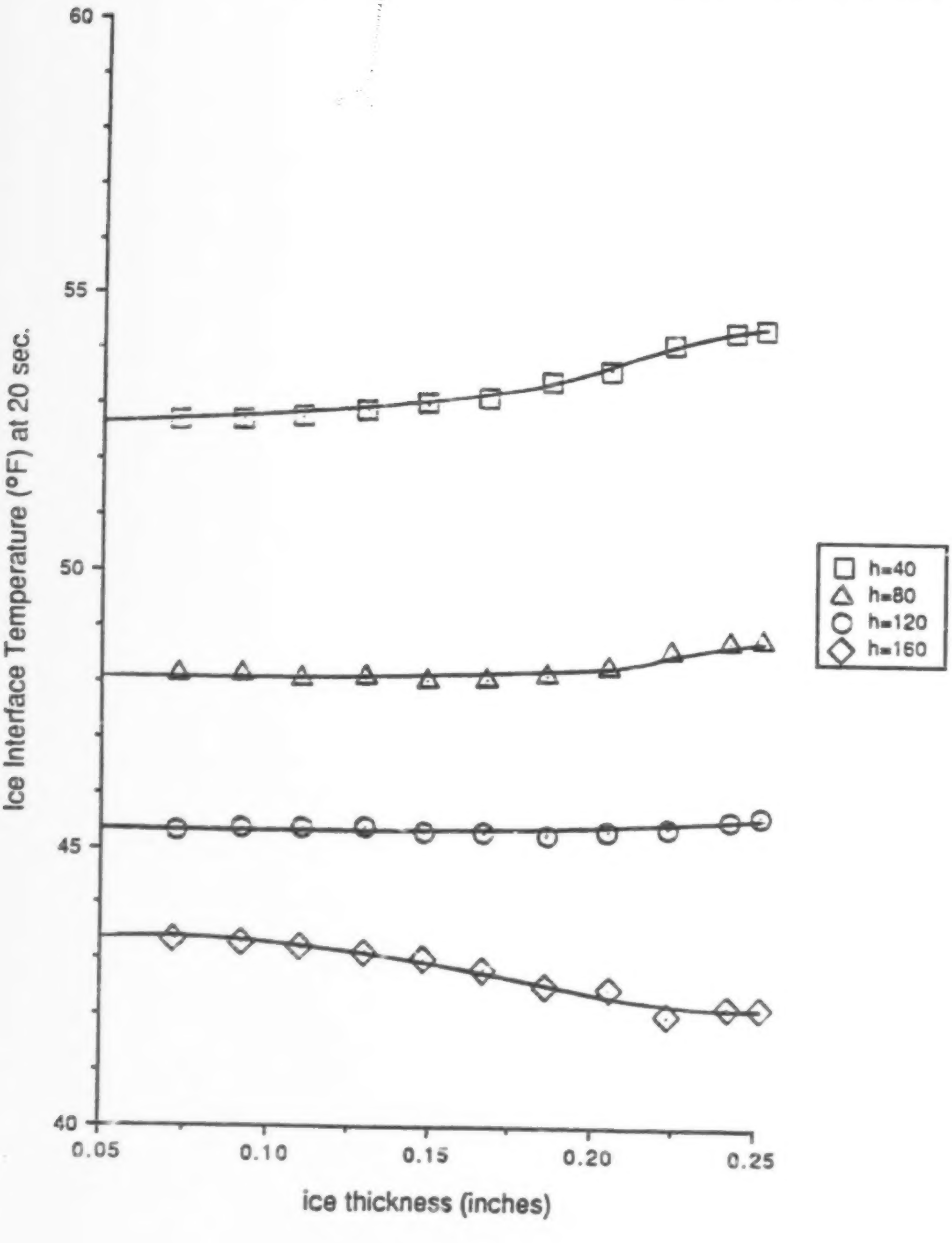
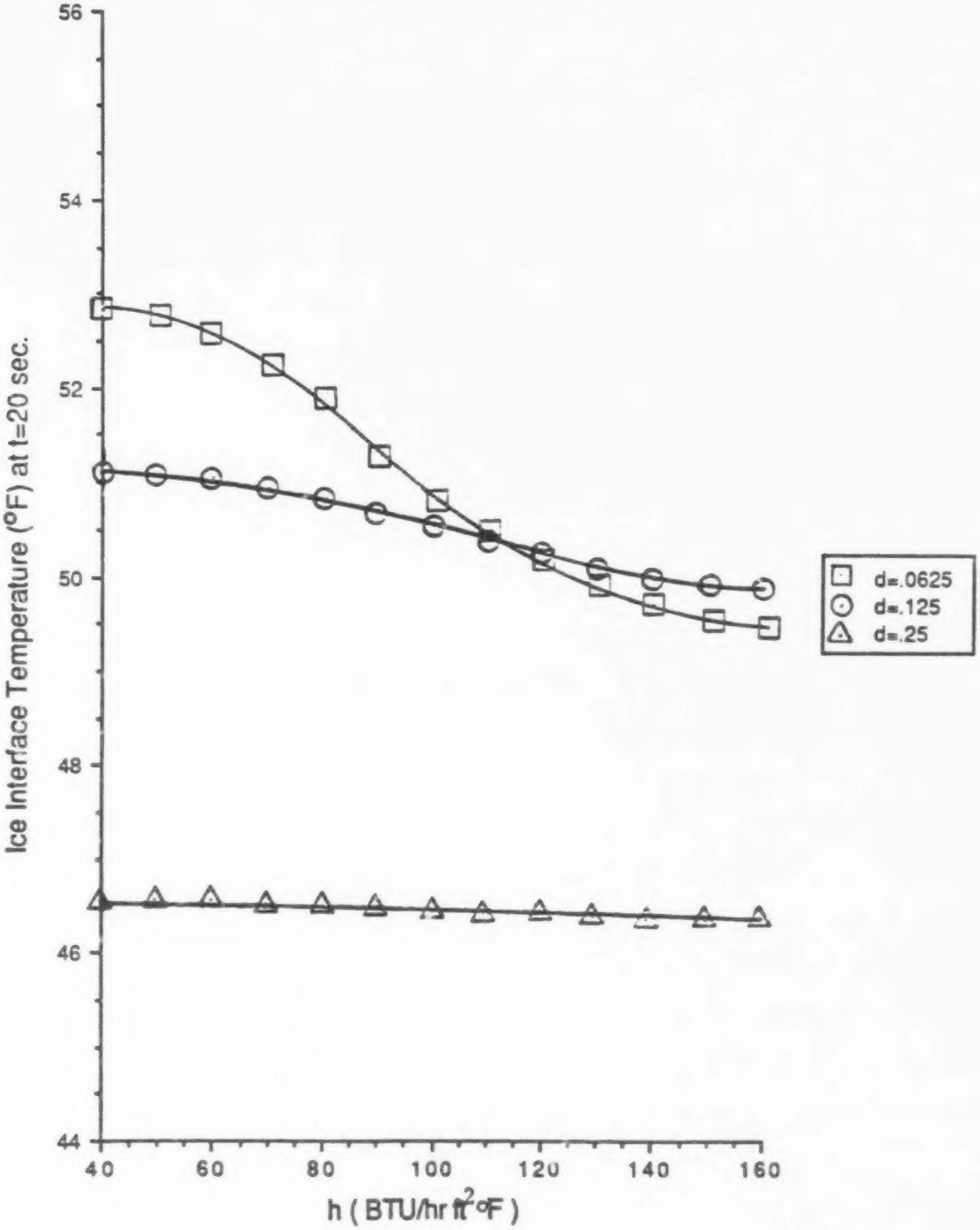


Figure 11

Effect of Variable Outer Heat Transfer Coefficient



Figures (10) and (11) show that the ice-abrasion shield interface temperature depends on both the ice thickness and the heat transfer coefficient when both are varying with position. However, the temperature change due to a variable ice shape is larger than that due to a variable heat transfer coefficient.

Figure (12) shows a plot of melt time versus ice thickness for the four heat transfer coefficients used in these cases and Fig. (13) shows melt time versus heat transfer coefficient for the four ice thicknesses run. The same points are being plotted in these two figures, but are plotted in two different ways for comparative purposes. Melt time is defined as the time needed for the temperature at the ice-abrasion shield interface to reach 32 °F. Figure (12) shows an interesting phenomena at very low ice thicknesses. A slight increase in melt time is noticed at a heat transfer coefficient of 160 BTU/hr -ft^2-°F and an ice thickness of 0.03125 inches. This point shows that for very thin ice, heat is being removed from the top surface by convection so fast that melting is taking longer to occur than it would for larger ice thicknesses. When the ice thickness becomes larger than a certain value, however, the convective surface is so distant that its effect on the abrasion shield temperature is less than the effect of the increasing ice thickness, and the melt times start to increase again. This suggests that there is a value of ice thickness at each heat transfer coefficient for which the melt time is a minimum.

In conclusion, for the effect of a variable heat transfer coefficient on the solution for the ice-abrasion shield interfacial temperature and for the melt times, it is felt that the heat transfer coefficient does not need to be variable except for cases which have sections of very thin ice (<0.125 in.), or no ice at all. In addition, the absolute value of the heat transfer coefficient does not have a strong effect on the interfacial temperature for ice thicknesses greater than 0.125 inches. For the effect of variable ice thickness on the interfacial temperature, it is felt that ice thickness can be modeled as a constant if the ice thickness at every point is at least 0.125 inches. It should be noted that the limiting ice thickess mentioned is dependent upon the heater power density. Smaller wattages will lower this value and higher wattages will raise it. Experience with using this code reveals that if the power density is one-half this value (15 W/in^2), then the limiting ice thickness is likewise cut in half. This ratio seems to hold for other power densities as well. This can be seen in the comparison with experimental data, which is discussed next.

Figure 12

Effect of Ice Thickness on Melt Times

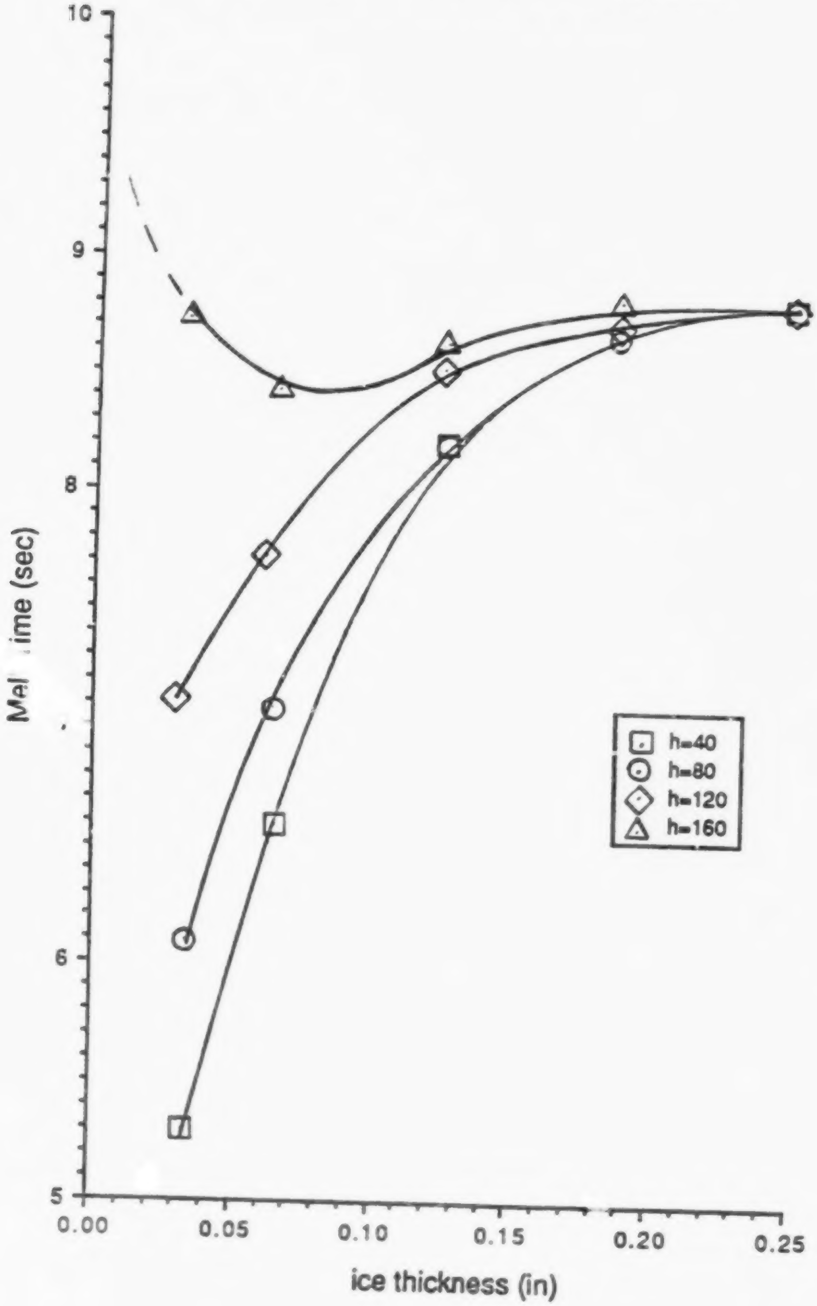
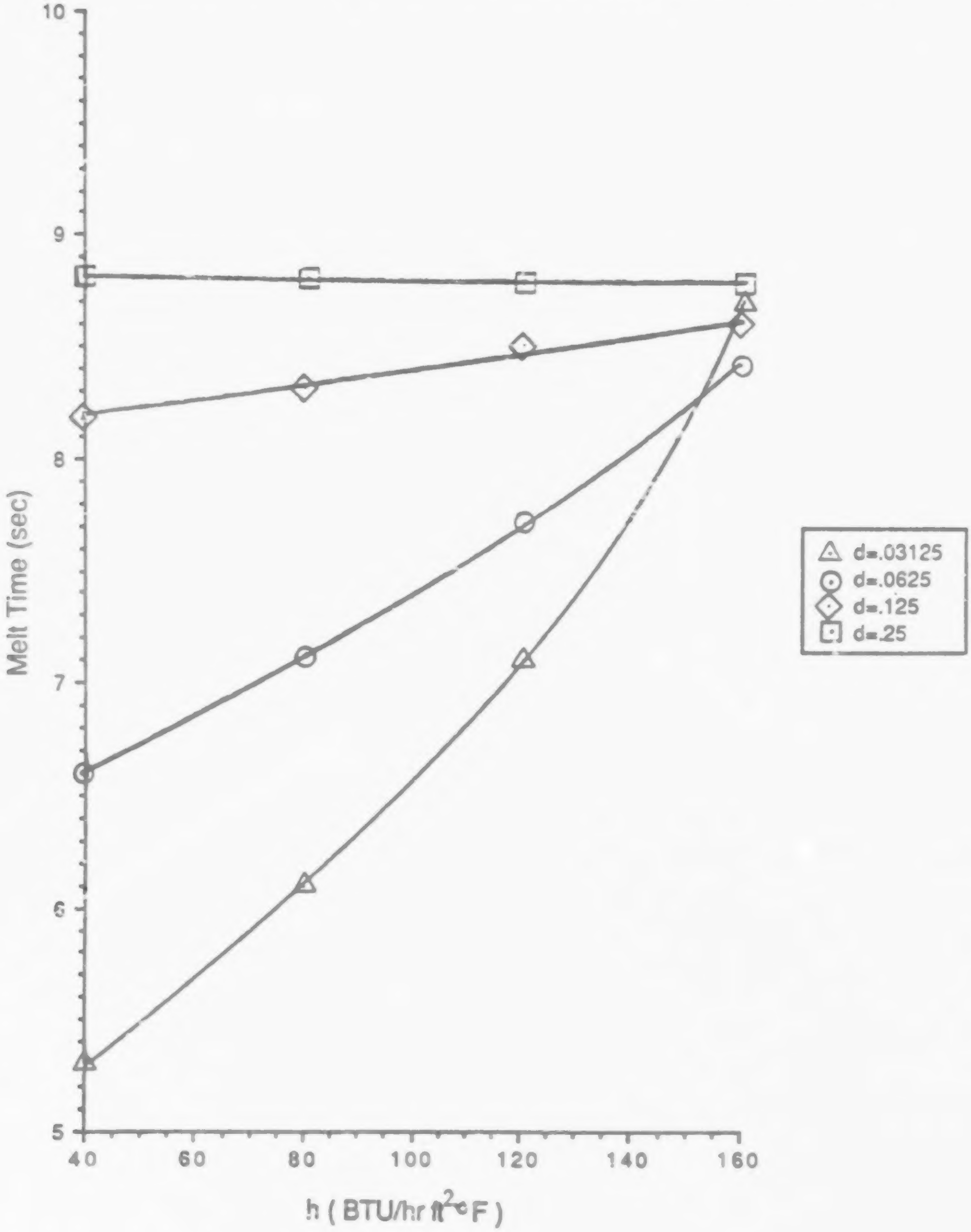


Figure 13

Effect of Outer Ambient Heat Transfer Coefficient on Melt Times



4. Comparison with Experimental Data

The computer program was run for comparison with previously obtained experimental data (7). The particular data set chosen was Run 305. It was selected because it was a case of cyclic heating. The ice shape and thickness were known from accretion data taken at the time of the experiment. The data for the deicer pad are given in Table

TABLE 6

Physical Properties of Experimental Data Case

Layer	Thickness(in)	Thermal Conductivity (BTU/hr-ft-°F)	Thermal Diffusivity (ft^2/hr)
Stainless Steel Abrasion Shield	0.0300	8.70	0.1500
Epoxy Adhesive	0.0168	0.10	0.0058
Epoxy/Glass Insulation	0.0138	0.22	0.0087
Epoxy Adhesive	0.0082	0.10	0.0058
Copper Heater Element	0.0065	60.00	1.1500
Epoxy Adhesive	0.0082	0.10	0.0058
Epoxy/Glass Insulation	0.0138	0.22	0.0087
Epoxy Adhesive	0.0082	0.10	0.0058
Stainless Steel Blade Skin	0.0200	8.70	0.1500
Epoxy Adhesive	0.0100	0.10	0.0058
Aluminum Doubler	0.0500	102.00	2.8300
Epoxy Adhesive	0.0100	0.10	0.0058
Aluminum D-Spar	0.1750	102.00	2.8300

6 and the various test conditions for this case are given in Table 7. Fourteen layers are modeled, which includes the ice layer. The actual blade used, Fig. (14), had different physical properties in the substrate at the stagnation point (Pos.6), but this can not be modeled using this code, and the properties at this position were assumed to be the same as for the rest of the composite blade. Six heaters are modeled, all of which are fired separately and have slightly different power densities. A case in which the heaters

Table 7

Test Conditions for Run 305

Heater Power Density Pos. 3 = 0.0 W/in^2
Heater Power Density Pos. 4 = 15.7 W/in^2
Heater Power Density Pos. 5 = 15.8 W/in^2
Heater Power Density Pos. 6 = 15.9 W/in^2
Heater Power Density Pos. 7 = 16.0 W/in^2
Heater Power Density Pos. 8 = 16.3 W/in^2

Pos. 4-6 are off the first 10 sec. of the run, on the next 10 sec., and then off again for the remainder of the run.

Pos. 7 is on for the first 10 sec. and off for the remainder of the run.

Pos. 8 is off for the first 20 sec., on for 10 sec., and off for the remainder of the run.

Variable ice shape is modeled using 0.0625 in. ice over Pos. 4,5, and 7 with no ice over Pos. 3,6, or 8.

Each heater element is 1 in. wide with a 0.061 in. gap between heaters.

Outer Convection Coefficient = 70. BTU/hr-ft^2-°F
Inner Convection Coefficient = 10. BTU/hr-ft^2-°F
Ambient Temperature = 20. °F

are not all fired at the same time was used to show two-dimensional effects which could not be simulated before since Chao's code contained only one heater. Where ice is present on the blade, it is modeled as having a constant value of 0.0625 inches due to the previous discussion on the effect of ice shape on the solution. Also, since the heater gap is very small in relation to the heater, a finer mesh is placed there in order to avoid possible accuracy problems. A constant heat transfer coefficient of 70 BTU/ft^2-hr-°F is assumed to be adequate for this simulation. A constant value can be assumed due to the previous discussion of the effect of heat transfer coefficient on the solution. Figure (14) shows the different positions of the heaters around the blade. Figures (15-20) plot temperature vs time at the ice-abrasion shield interface, the bottom of the heater and the substrate for both the experimental and numerical results. The experimental results are shown as a solid line due to the fact that this is the manner in which the graphics routine plots the data. The three points plotted at each position reflect the position of the thermocouples at each position. Beyond the six positions given here, the experimental results agree with Marano's (3) one-dimensional code and therefore were not modeled with this code.

As can be seen from these figures, excellent agreement with the experimental data is obtained at positions 5, 7, and 8. At positions where ice is present, melting can be observed in both the numerical and experimental results by a slight change in slope which can be seen in the ice-abrasion shield interface temperatures.This can be seen more clearly by comparing the response at Position 6, which has no ice, with the response at Position 5 which does have ice. There is a change of slope at 32 °F at Position 5 which indicates the ice is melting. At Position 6, the abrasion shield temperature exceeds 40 °F without any change in slope, which indicates that ice is not present at this position.

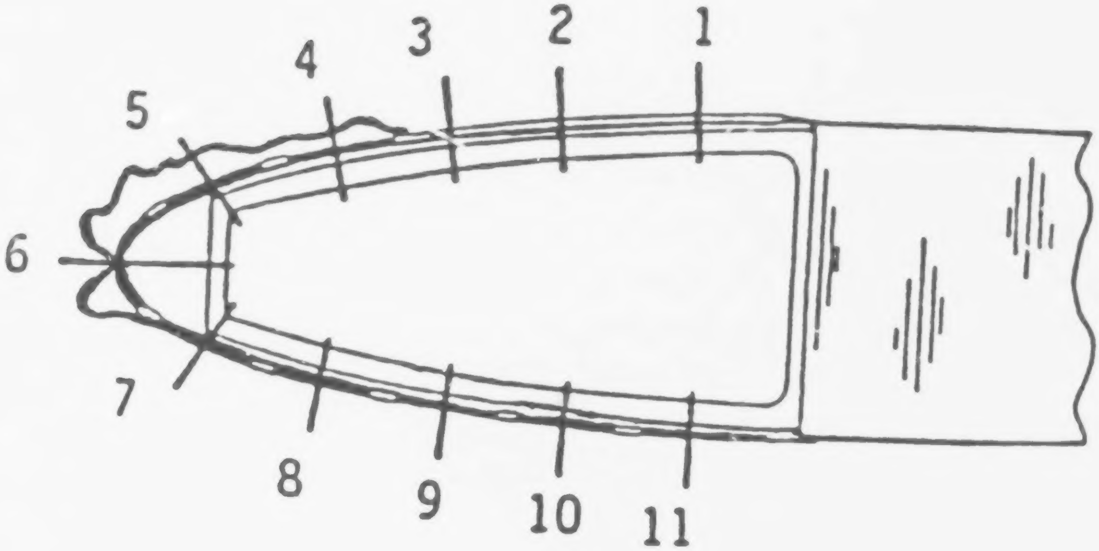
The response given at Position 3 shows a large amount of transverse heating since a significant temperature response is noted even though this position does not contain a heater. The numerical results also show a response at this position, although the simulation under-predicts the experimental values. As can be seen from the experimental results, however, the temperature of the abrasion shield at Position 3 is higher than the corresponding value at Position 4, which does have a heater. Since the heat being received at Position 3 comes from Position 4, it does not seem to be possible for the response to be higher at Position 3 than at Position 4.

The maximum heater temperatures which are predicted at Positions 4 and 6 are also lower than the experimental results despite the fact that the numerical simulation predicts the substrate and abrasion shield temperatures at these positions very well. However, the program predicts the maximum heater temperatures at the other positions adequately even though these positions have roughly the same power density as Positions 4 and 6. It is believed that air gaps near the heaters at Positions 4 and 6 are responsible for the difference in heater temperature. An air gap would cause the heater to reach a much higher temperature, but would not significantly change the response at other locations. This is, in fact, what is seen at these positions.

This case is clearly two-dimensional in this region owing to the different cycles on the heaters and to the absence of ice at some points on the blade. Any case which has heaters firing in different cycles should be modeled with a two-dimensional code since a one-dimensional code can not account for the transverse heating which results from sequential firing of the heaters. Transverse heating is also very evident at Position 3, which does not have a heater. Similarly, a one-dimensional code can not handle having an ice shape such as the one present in this experimental case. However, the heater gap in the experimental case is not large enough to warrant a two-dimensional analysis, as this does not create much of a two-dimensional effect. This can be seen from the fact that Positions 9-11 are adequately modeled using an one-dimensional analysis even though these positions contain a heater gap, but display no other two-dimensional behavior.

Figure 14

Positions of Heaters Around the Blade



1
2
3
4
5
6
7
8
9
10
11

Figure 15

Comparison of Numerical Model to Experimental Data: Heater Position 3

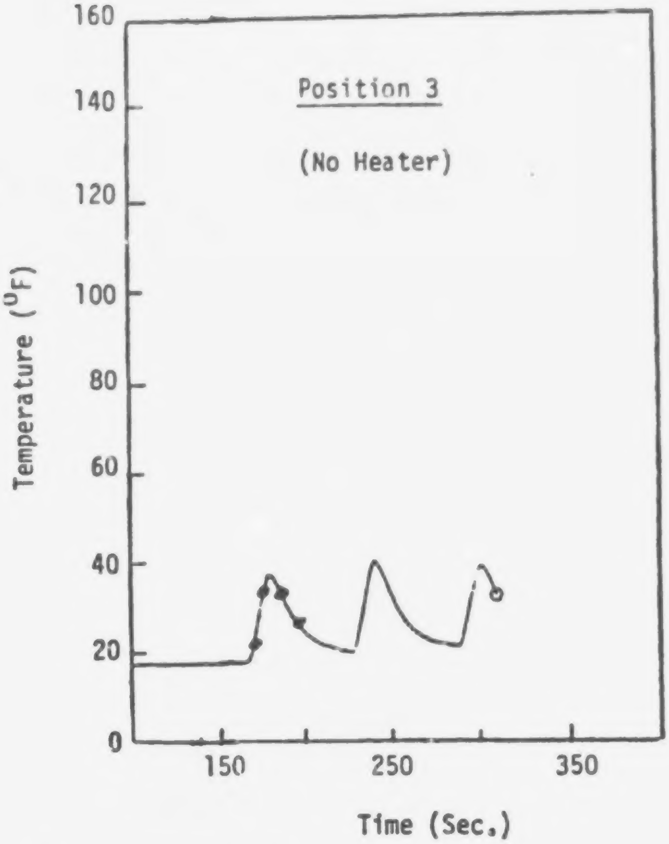


Figure 16

Comparison of Numerical Model to Experimental Data: Heater Position 4

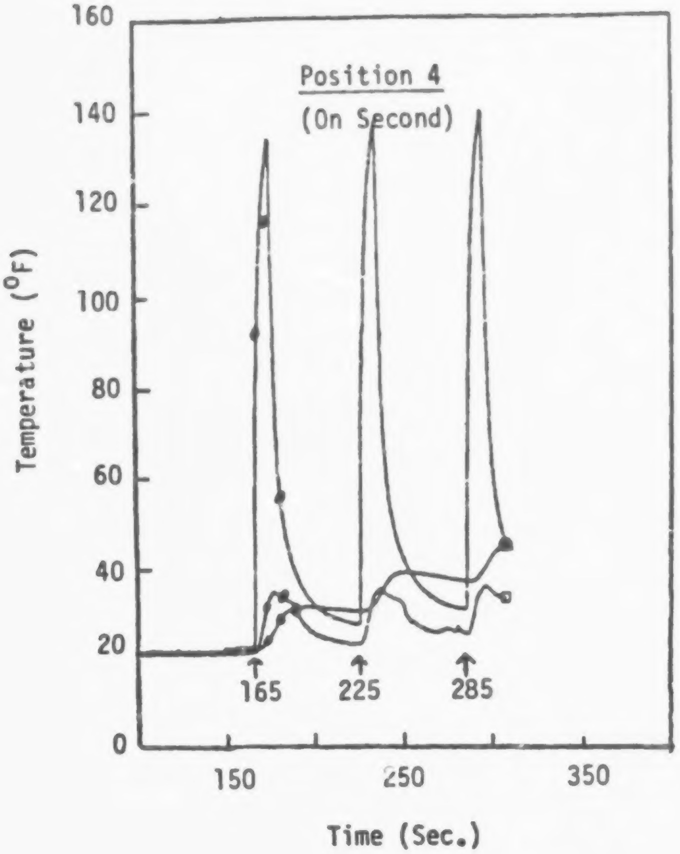


Figure 17

Comparison of Numerical Model to Experimental Data: Heater Position 5

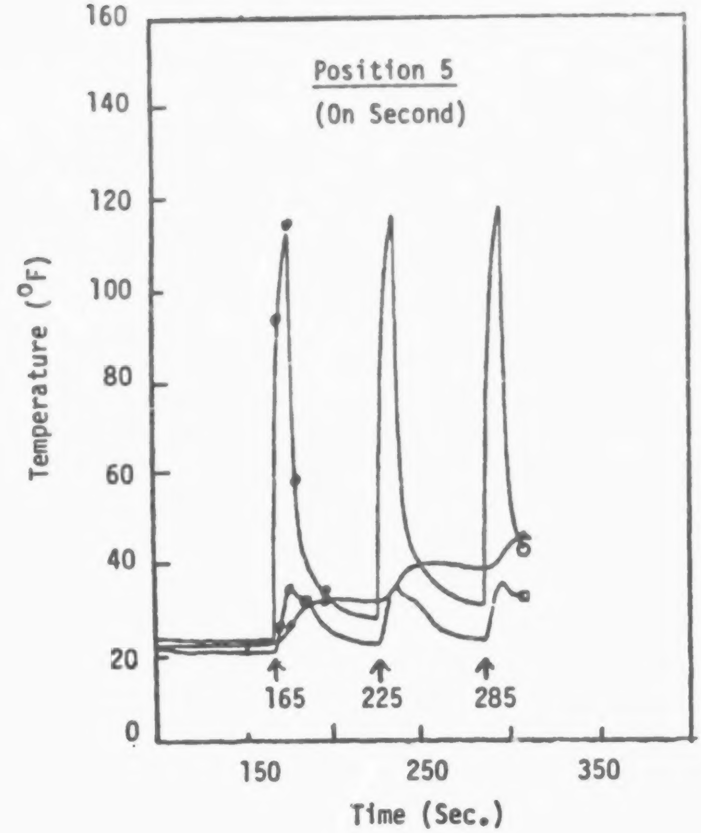


Figure 18

Comparison of Numerical Model to Experimental Data: Heater Position 6

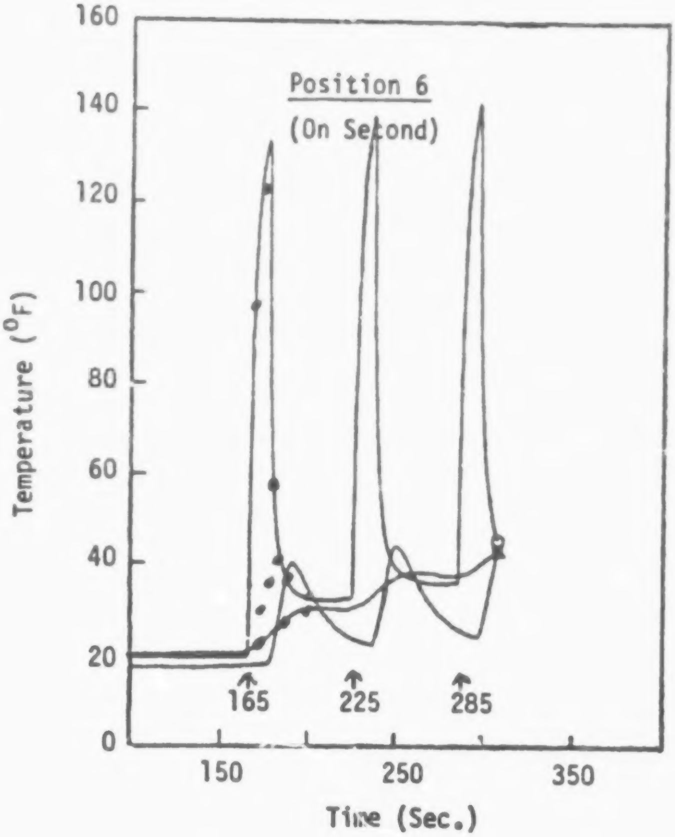


Figure 19

Comparison of Numerical Model to Experimental Data: Heater Position 7

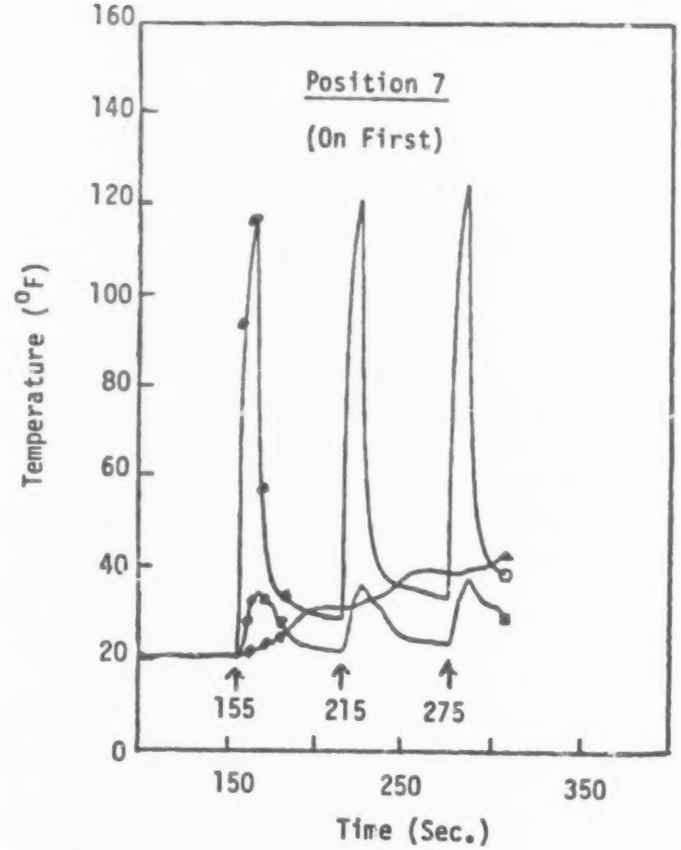
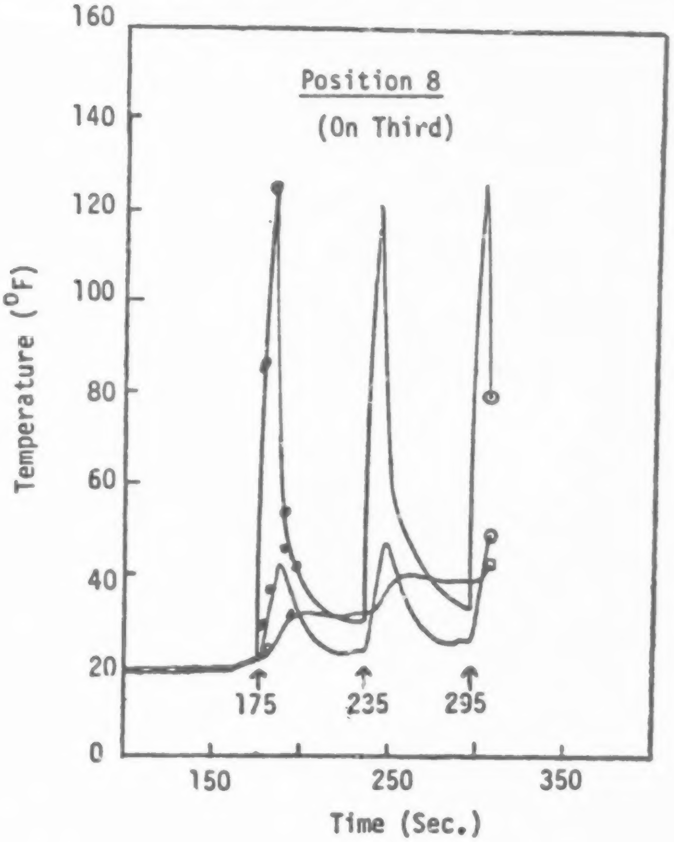


Figure 20

Comparison of Numerical Model to Experimental Data: Heater Position 8



VII. CONCLUSIONS AND RECOMMENDATIONS

The computer code developed in this study for the two-dimensional heat transfer in an electrothermal deicer pad has been shown to predict accurate results for seven different numerical methods and for three different solution techniques. An analysis was performed to determine the most efficient method for this problem. Parametric studies were performed to investigate the effect of gap width, nodal spacing, ice thickness, and outer heat transfer coefficient. An analysis was performed to determine when a two-dimensional model is preferred over a one-dimensional model. Finally, a comparison was made with existing experimental data. The results of the numerical simulation were found to compare very well with previous numerical calculations, as well as with the experimental data.

An additional comparison was made with an experimental data case supplied by ONERA. This data was not supplied in time to be included in this thesis, but the results were found to agree very well with this data and with those obtained via codes developed by ONERA and RAE.

The computer program was run on the University of Toledo NAS9080 computer. A 12 x 48 mesh was used to model a 6 layer composite body with one heater. Typical computing time for a simulation time of 20 seconds was 15 seconds for the ADI method, as compared to 4 minutes 35 seconds for the program which was used prior to this study.

The analysis performed shows that a two-dimensional model is preferred over a one-dimensional model when the parameters discussed in the results section exceed certain values or when the computational grid is small enough so that the extra computer time used by the two-dimensional model is minimal.

The simulation contains the following improvements over the existing two-dimensional numerical technique for an electrothermal deicer pad:

(1) The Method of Assumed States was used to modify the Enthalpy method so that more efficient numerical techniques could be implemented;

(2) The Alternating Direction Implict (ADI) method was implemented and was found to be the most efficient of the numerical methods studied;

(3) The simulation can model more complex two-dimensional cases such as a variable number of heaters, variable firing sequences of these heaters, more accurate modeling of the gap between heaters, variable ice growth, and a variable outer heat transfer coefficient.

Based on the progress thus far, it is recommended that the following additional work be carried out:

(1) Additional experimental work needs to be performed with cyclic heaters to fully understand the two-dimensional behavior of a deicer and to further verify the numerical codes;

(2) A two-dimensional deicer code which models the true shape of the blade has recently been developed, but needs a similar analysis to the one performed here to decrease its computer time;

(3) The code developed here, or the one mentioned above, needs to be combined with accretion codes, trajectory codes, and other numerical codes to obtain a code which models the total deicing problem.

VIII. REFERENCES

1. Stallabrass, J. R., "Thermal Aspects of Deicer Design", presented at the International Helicopter Icing Conference, Ottawa, Canada, 1972.

2. Baliga, G., "Numerical Simulation of One-Dimensional Heat Transfer in Composite Bodies with Phase Change", M. Sc. Thesis, University of Toledo, Toledo, Ohio, 1980.

3. Marano, J. J., "Numerical Simulation of an Electrothermal Deicer Pad", M.Sc. Thesis, University of Toledo, Toledo, Ohio, May 1982.

4. Gent, R. W. and J. T. Cansdale, "One-Dimensional Treatment of Thermal Transients in Electrically Deiced Helicopter Rotor Blades", RAE Technical Report 80159, 1980.

5. Roelke, R. J., "A Rapid Computational Procedure for the Numerical Solution of a Heat Flow Problem with Phase Change", M. Sc. Thesis, University of Toledo, Toledo, Ohio, August 1986.

6. Chao, D. F., "Numerical Simulation of Two-Dimensional Heat Transfer in Composite Bodies with Application to Deicing of Aircraft Components", PhD Thesis, University of Toledo, Toledo, Ohio, Nov. 1983.

7. Leffel, K. L., "A Numerical and Experimental Investigation of Electrothermal Aircraft Deicing", M. Sc. Thesis, University of Toledo, Toledo, Ohio, Jan. 1986.

8. Ockenden, J. R. and W. R. Hodgkins (editors), *Moving Boundary Value Problems in Heat Flow and Diffusion*, Oxford Univ. Press, Oxford, 1975.

9. Bonacina, C., G. Comini, A. Fasano, and N. Primicerio, "Numerical Solution of Phase Change Problems", *Int. J. Heat & Mass Trans.*, vol. 10, p.1825, 1973.

10. Voller, V. and M. Cross, "Accurate Solutions of Moving Boundary Value Problems Using the Enthalpy Method", *Int. J. Heat & Mass Trans.*, vol. 24, p.545, 1981.

11. Voller, V., M. Cross and P. G. Walton, "Assessment of Weak Solution Numerical Techniques for Solving Stefan Problems', 172, Dept. of Mathematics & Computer Studies, Sunderland Polytechnic, U. K., 1979.

12. Atthey, D. R., "A Finite Difference Scheme for Melting Problems", *J. Inst. Math. Appl.*, vol. 13, p. 353, 1974.

13. Schneider, G. E. and M. J. Raw, "An Implicit Solution Procedure for Finite Difference Modeling of the Stefan Problem", *AIAA Journal* , vol. 22, Nov. 1984.

14. Carslaw, H. S. and J. C. Jaeger, *Conduction of Heat in Solids*, Clarendon Press, Oxford, 1959

15. Carnahan, B., H. A. Luther and J. O. Wilkes, *Applied Numerical Methods*, Wiley, New York, 1969.

16. von Rosenburg, D. W., *Modern Analytical and Computational Methods in Science and Mathematics*, American Elsevier, 1969.

17. Anderson, D. A., J. C. Tannehill and R. H. Pletcher, *Computational Fluid Dynamics and Heat Transfer*, McGraw-Hill, New York, 1984.

18. Barakat, H. Z. and J. A. Clark, "On the Solution of the Diffusion Equations by Numerical Methods", *J. Heat Trans.*, pp. 421-427, Nov. 1966.

19. Peaceman, D. W. and H. H. Rachford, Jr., "The Numerical Solution of Parabolic and Elliptic Differential Equations", *J. Soc. Indust. Appl. Math.*, vol. 3, no. 1, March 1955.

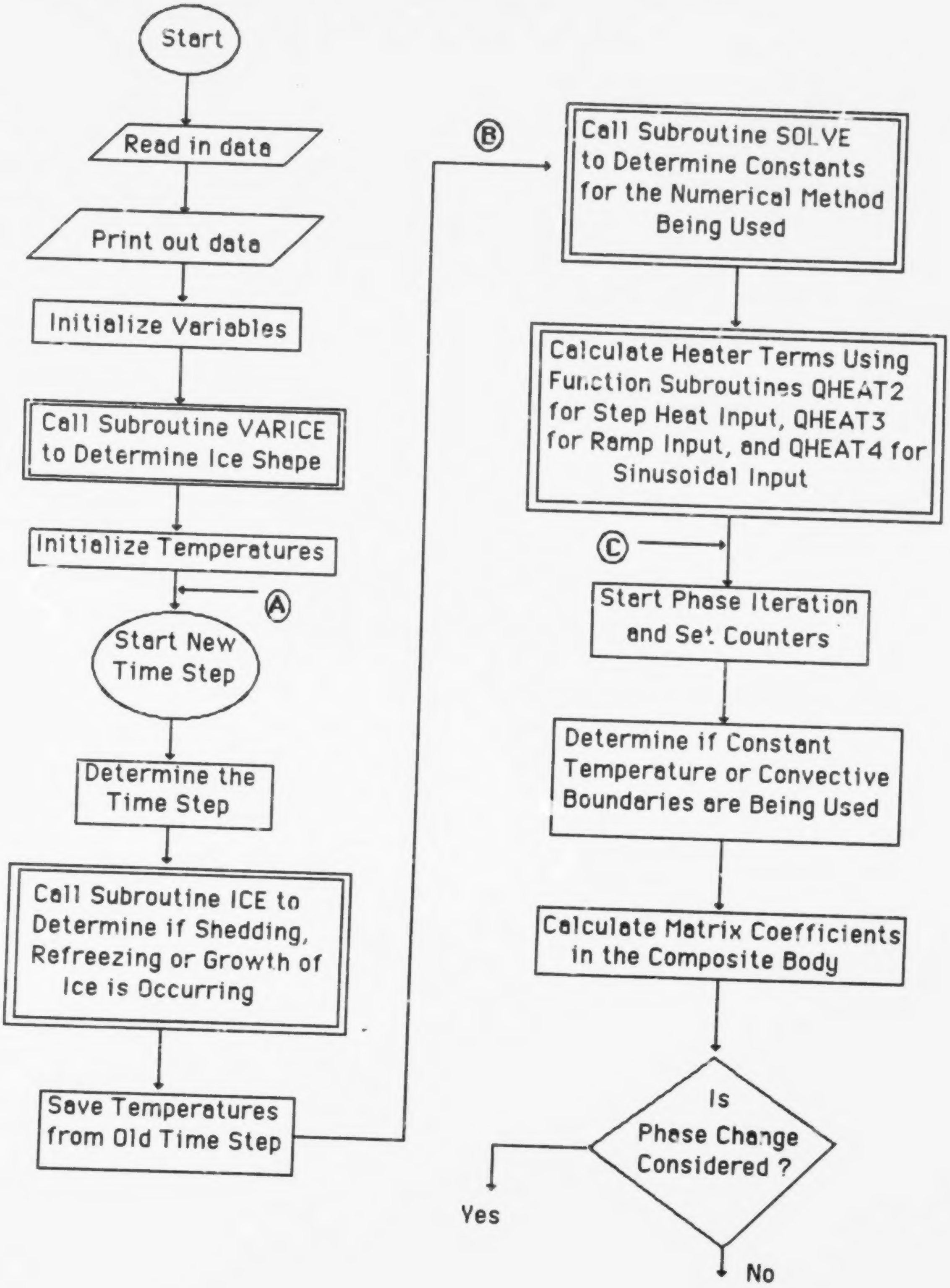
20. Yanenko, N. N. *The Method of Fractional Steps: The Solution of Problems of Mathematical Physics in Several Variables*, (M. Holt, ed.), Springer-Verlag, New York.

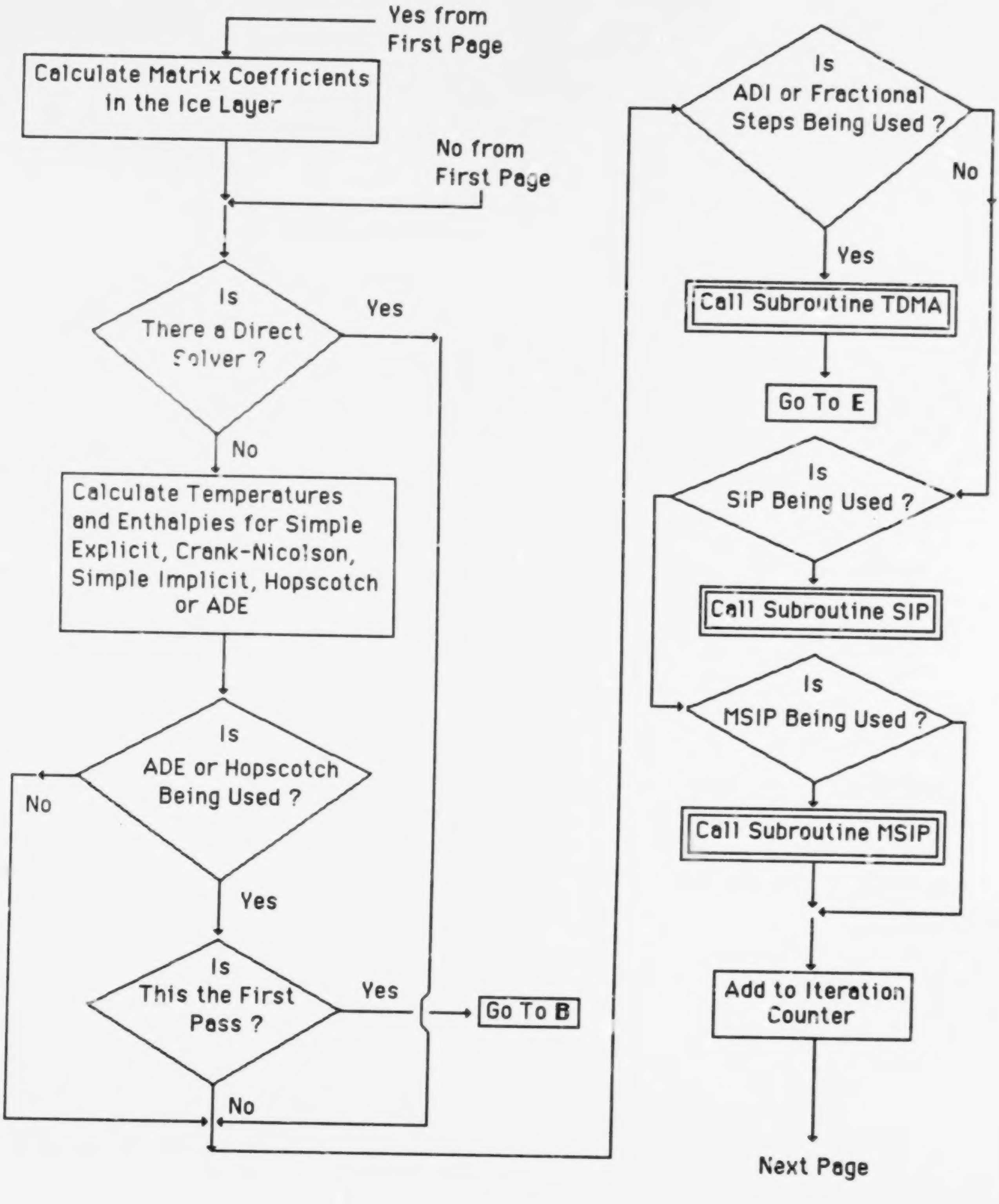
21. Stone, H. L., "Iterative Solution of Implicit Approximation of Multidimensional Partial Differential Equations", *SIAM Journal of Numerical Analysis*, vol. 5, pp. 530-558, Sept. 1968.

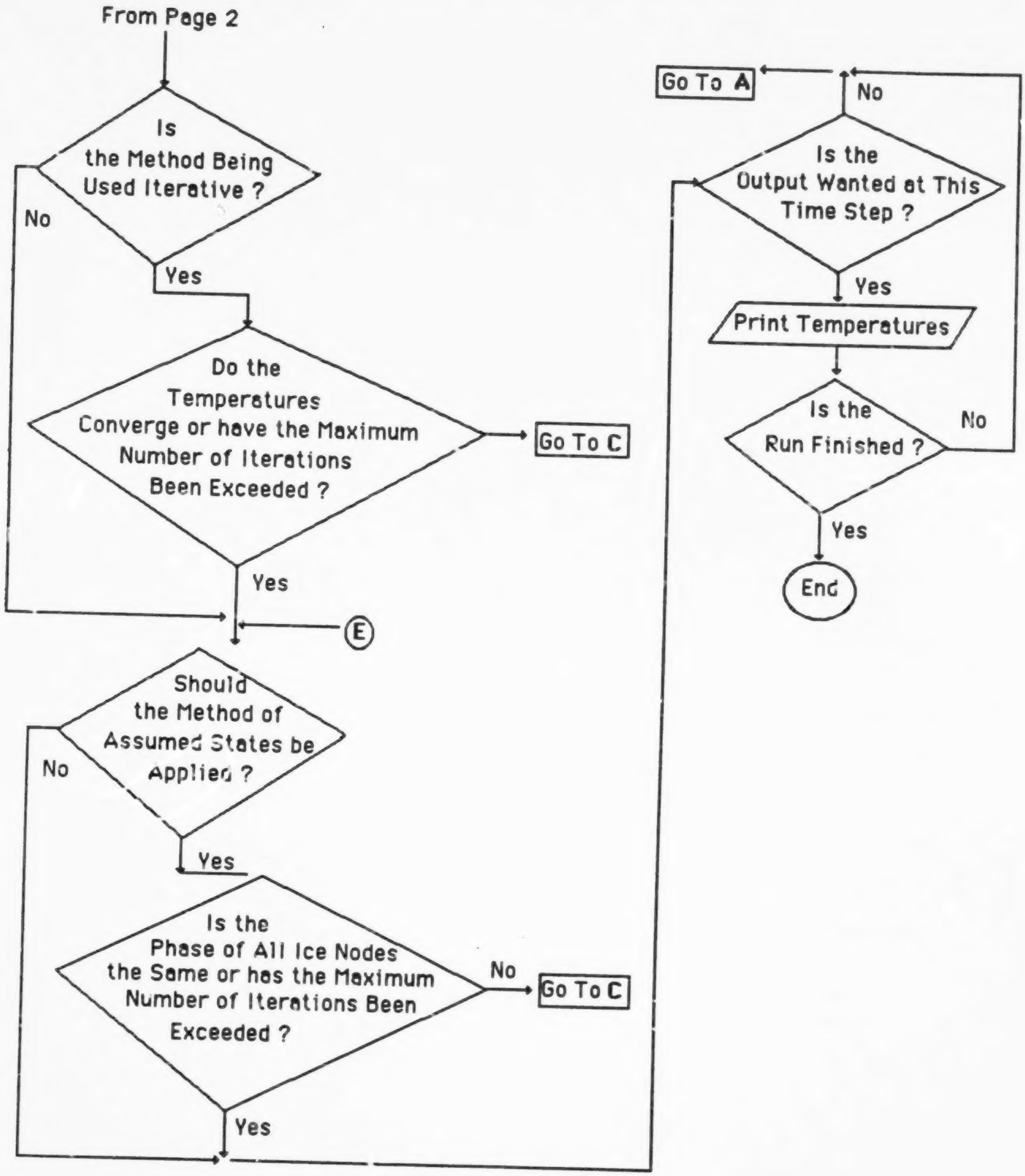
22. Schneider, G. E. and M. Zedan, "A Modified Strongly Implicit Procedure for the Numerical Solution of Field Problems", *Numerical Heat Transfer*, vol. 4, pp. 1-19, 1981.

APPENDIX: Flow Diagram, Program Listing and Sample Input Data

Flowsheet of Program


Start
Read in data
Print out data
Initialize Variables
Call Subroutine VARICE to Determine Ice Shape
Initialize Temperatures
Ⓐ
Start New Time Step
Determine the Time Step
Call Subroutine ICE to Determine if Shedding, Refreezing or Growth of Ice is Occurring
Save Temperatures from Old Time Step
Ⓑ
Call Subroutine SOLVE to Determine Constants for the Numerical Method Being Used
Calculate Heater Terms Using Function Subroutines QHEAT2 for Step Heat Input, QHEAT3 for Ramp Input, and QHEAT4 for Sinusoidal Input
Ⓒ
Start Phase Iteration and Set Counters
Determine if Constant Temperature or Convective Boundaries are Being Used
Calculate Matrix Coefficients in the Composite Body
Is Phase Change Considered ?
Yes
No


Yes from First Page
Calculate Matrix Coefficients in the Ice Layer
No from First Page
Is There a Direct Solver ?
Yes
No
Calculate Temperatures and Enthalpies for Simple Explicit, Crank-Nicolson, Simple Implicit, Hopscotch or ADE
Is ADE or Hopscotch Being Used ?
No
Yes
Is This the First Pass ?
Yes
Go To B
No
Is ADI or Fractional Steps Being Used ?
No
Yes
Call Subroutine TDMA
Go To E
Is SIP Being Used ?
Call Subroutine SIP
Is MSIP Being Used ?
Call Subroutine MSIP
Add to Iteration Counter
Next Page


From Page 2
Is the Method Being Used Iterative ?
No
Yes
Do the Temperatures Converge or have the Maximum Number of Iterations Been Exceeded ?
Go To C
Yes
E
Should the Method of Assumed States be Applied ?
No
Yes
Is the Phase of All Ice Nodes the Same or has the Maximum Number of Iterations Been Exceeded ?
No
Go To C
Yes
Go To A
No
Is the Output Wanted at This Time Step ?
Yes
Print Temperatures
Is the Run Finished ?
No
Yes
End

```
//UOFT1572 JOB (UT,,,
// 06200016,1,,,V)
// EXEC   FORTVCLG,TIME.GO=60,PARM='NOSOURCE,NOSRCFLG'
//FORT.SYSIN  DD *
C
C        NUMERICAL SIMULATION OF TWO-DIMENSIONAL HEAT TRANSFER
C           IN COMPOSITE BODIES WITH PHASE CHANGE.
C
C        THIS PROGRAM CAN CALCULATE THE TEMPERATURE PROFILE IN
C        A COMPOSITE SLAB WHICH HAS CONVECTIVE,CONSTANT TEMP-
C        ERATURE OR MIXED BOUNDARY CONDITIONS.
C
C        THE PROGRAM CAN ALSO BE USED FOR COMPOSITE BODY PROBLEMS
C              WITH CONSTANT OR VARIABLE HEAT SOURCES.
C
C        IT ALSO HAS OPTIONS FOR VARIABLE ICE SHAPES AND VARIABLE
C        HEAT TRANSFER COEFFICIENTS AT THE TOP SURFACE.
C
C
C             IT CAN ALSO PERFORM THESE CALCULATIONS USING
C             ANY ONE OF EIGHT DIFERENT NUMERICAL METHODS.
C
C
C        COMPLETED 6/1/87 AS PARTIAL FULFILLMENT FOR A MASTER'S THESIS
C        IN CHEMICAL ENGINEERING AT THE UNIVERSITY OF TOLEDO.
C
C                     AUTHOR: WILLIAM B. WRIGHT
C
C        AK,AKX,AAK    = THERMAL CONDUCTIVITIES OF LAYERS.
C        ALP           = THERMAL DIFFUSIVITIES OF LAYERS.
C        AKL,AKS       = THERMAL CONDUCTIVITY OF WATER AND ICE.
C        ALPL,ALPS     = THERMAL DIFFUSIVITY OF WATER AND ICE.
C        CHANGE        = SUBROUTINE TO CORRECTING THERMAL CONDUCTIVITIES
C                        OF EACH NODE IN THE ICE LAYER FOR PHASE CHANGE
C        CPS,CPL       = SPECIFIC HEAT OF ICE AND WATER PER UNIT VOLUME.
C        DES,DEL       = DENSITY OF ICE AND WATER.
C        DTAUI         = INITIAL TIME STEP.
C        DTAUM         = INTERMEDIATE TIME STEP.
C        DTAUF         = FINAL TIME STEP.
C        DX            = SPACING BETWEEN NODES IN THE X-DIRECTION OF THE
C                        GRID.
C        DY            = SPACING BETWEEN NODES IN THE Y-DIRECTION OF THE
C                        GRID.
C        EL            = LENGTH OF EACH LAYER IN THE Y-DIRECTION
C        ELX           = LENGTH OF EACH LAYER IN THE X-DIRECTION
C        EPSI          = CONVERGENCE CRITERIA FOR ITERATION
C        H,HO          = ENTHALPY OF NODE AT PRESENT AND PREVIOUS TIME
C                        STEPS.
C        HEAD          = HEADINGS
C        H1,H2,H3,H4   = HEAT TRANSFER COEFF. AT LOWER,UPPER, AND SIDE
C                        BOUNDARIES.
C        HLAM          = LATENT HEAT OF ICE.
C        IBC1,IBC2     = 1, IMPLIES TEMPERATURE IS CONSTANT AT LOWER,
C        IBC3,IBC4          UPPER, AND SIDE BOUNDARIES.
```

```
C
C        IBC1,IBC2      = 2, IMPLIES CONVECTIVE HEAT TRANSFER AT LOWER,
C        IBC3,IBC4           UPPER, AND SIDE BOUNDARIES.
C
C        ICE            = SUBROUTINE FOR SHEDDING, REFREEZING, AND GROWTH
C                         OF ICE
C        ICOUNT         = COUNTER ON TIME STEP.
C        IFREQ          = NUMBER OF TIME STEPS BETWEEN SUCCESSIVE
C                                 PRINTING OF THE TEMPERATURE PROFILE.
C        IG             = 1, IMPLIES PHASE CHANGE IN ICE LAYER IS NOT
C                            CONSIDERED. ISH AND IRF MUST BE SET EQUAL
C                            TO ONE FOR THIS CASE.
C        IG             = 2, IMPLIES PHASE CHANGE IN ICE LAYER IS
C                                                         CONSIDERED.
C        IH             = 1, IMPLIES NO HEAT SOURCE.
C        IH             = 2, IMPLIES POINT HEAT SOURCE.
C        IH             = 3, IMPLIES HEAT GENERATION WITHIN SLAB.
C        IHQ            = 1, IMPLIES CONSTANT HEAT SOURCE.
C        IHQ            = 2, IMPLIES STEP FUNCTION HEAT SOURCE.
C        IHQ            = 3,  IMPLIES RAMP FUNCTION HEAT SOURCE.
C        IHQ            = 4,  IMPLIES SINE FUNCTION HEAT SOURCE.
C        IJ             = SLAB WITHIN WHICH HEAT GENERATION OCCURS.
C        IRF            = 1, IMPLIES REFREEZING ICE LAYER IS NOT
C                                                     CONSIDERED.
C        IRF            = 2, IMPLIES REFREEZING ICE LAYER IS CONSIDERED.
C        ISH            = 1, IMPLIES SHEDDING ICE LAYER IS NOT CONSIDERED.
C                         IRF MUST ALSO BE ONE FOR THIS CASE.
C        ISH            = 2, IMPLIES SHEDDING ICE LAYER IS CONSIDERED.
C        ITYPE          = DEFINES THE SHAPE OF THE VARIABLE ICE SHAPE.
C                       = 1, ICE NODE IS IN THE INTERIOR OF THE ICE LAYER.
C                       = 2, CONVECTIVE BOUNDARY ON TOP OF ICE NODE ONLY.
C                       = 3,     ''        ''          LEFT ''    ''
C                       = 4,     ''        ''          RIGHT ''   ''
C                       = 5,     ''        ''   TOP AND LEFT ''    ''
C                       = 6,     ''        ''   TOP AND RIGHT ''   ''
C                       = 7,     ''        '' LEFT AND RIGHT ''   ''
C                       = 8,     ''       TOP, LEFT AND RIGHT ''   ''
C                       = 9, ICE NODE IS AT ICE-ABRASION SHIELD INTERFACE
C
C        JCOUNT         = MAXIMUM NUMBER OF ITERATIONS FOR TIME STEP.
C        KCOUNT         = COUNTER ON PHASE ITERATION
C        KSH            = NODE AT WHICH SHEDDING BEGINS
C        L              = NUMBER OF LAYERS IN SLAB
C        L1             = LOWER SLAB NUMBER FOR POINT HEAT SOURCE.
C        L2             = UPPER SLAB NUMBER FOR POINT HEAT SOURCE.
C        M              = NUMBER OF NODES IN THE Y-DIRECTION OF THE GRID
C        METHOD         = FINITE DIFFERENCING METHOD USED IN SIMULATION
C                       = 1, COMBINED METHOD(SEE DESCRIPTION OF THETA)
C                       = 2, HOPSCOTCH
C                       = 3, ADE (ALTERNATING-DIRECTION EXPLICIT)
C                       = 4, ADI (ALTERNATING-DIRECTION IMPLICIT)
C                       = 5, TIME-SPLITTING METHOD(THETA > OR = .5)
C                       = 6, SIP (STRONGLY IMPLICIT PROCEDURE)
C                       = 7, MSIP (MODIFIED STRONGLY IMPLICIT PROCEDURE)
```

```
C                        THETA > OR = .5 FOR SIP AND MSIP
C
C     MM             = INTERFACE NODE NUMBERS IN THE Y-DIRECTION
C     MSIP           = SUBROUTINE WHICH SOLVES FOR TEMP. USING MSIP
C     MX             = INTERFACE NODE NUMBERS IN THE X-DIRECTION
C     N              = NUMBER OF NODES IN THE X-DIRECTION OF THE GRID
C     NISP           = NUMBER OF TIME STEPS FOR WHICH INTIAL TIME
C                                                      STEP IS USED.
C     NMSP           = NUMBER OF TIME STEPS FOR WHICH INTERMIDIATE
C                                                  TIME STEP IS USED.
C     NFSP           = NUMBER OF TIME STEPS FOR REFREEZING ICE LAYER.
C                      EQUAL TO ZERO IF REFREEZING IS NOT CONSIDERED.
C     NHSP           = NUMBER OF TIME STEPS FOR SHEDDING AT 2ND CYCLE.
C                      EQUAL TO ZERO IF SHEDDING IS NOT CONSIDERED.
C     NTSP           = NUMBER OF TIME STEPS FOR STOPPING THE PROGRAM.
C     NO             = NUMBER OF NODES IN EACH LAYER IN THE Y-DIRECTION
C                      OF THE GRID
C     NOX            = NUMBER OF NODES IN EACH LAYER IN THE X-DIRECTION
C                      OF THE GRID
C     NO1            = LOWER NODE NUMBER FOR FINITE THICKNESS HEATER.
C     NO2            = UPPER NODE NUMBER FOR FINITE THICKNESS HEATER.
C     NODE           = NODE AT WHICH POINT HEAT SOURCE IS APPLIED.
C     PA             = DENSITY*HEAT CAPACITY AT ANY POINT I,J
C     PHASE          = SUBROUTINE USED TO DETERMINE PHYSICAL PROPERTIES
C                      IN THE ICE LAYER
C     Q              = HEAT SOURCE  WATTS/IN*IN.
C     QHEAT2         = FUNCTION PROGRAM FOR STEP FUNCTION HEAT INPUT.
C     QHEAT3         = FUNCTION PROGRAM FOR RAMP FUNCTION HEAT INPUT.
C     QHEAT4         = FUNCTION PROGRAM FOR SINE FUNCTION HEAT INPUT.
C     QV             = INPUT FOR VARIABLE HEAT SOURCE.
C     RATE           = RATE OF ICE GROWTH(IN/S) AS A FUNCTION OF X
C     SIP            = SUBROUTINE WHICH SOLVES FOR TEMP. USING SIP
C     SOLVE          = SUBROUTINE USED TO DEFINE VALUES FOR THR, THT,
C                      THL, AND THB
C     T,TO           = NON-DIMENSIONAL TEMPERATURES AT PRESENT AND
C                      PREVIOUS TIME STEPS.
C     TDMA           = SUBROUTINE WHICH SOLVES FOR TEMP. USING TDMA
C     TE             = TEMPERATURE.
C     TG1,TG2        = AMBIENT TEMPERATURE AT LOWER ,UPPER, AND SIDE
C     TG3,TG4          BOUNDARIES OF SLAB.
C     THR,THL        = DETERMINES FINITE-DIFFERENCE METHOD
C     THT,THB          IF ALL ARE 0.0 , EXPLICIT
C                      IF ALL ARE 0.5 , CRANK-NICOLSON
C                      IF ALL ARE 1.0 , TOTALLY IMPLICIT
C                      COMBINATION OF THESE ARE USED FOR OTHER METHODS
C
C     TIN            = INITIAL TEMPERATURE IN SLAB.
C     TLAG           = LAG TIME BEFORE HEATER COMES ON FOR VARIABLE
C                      HEAT INPUT
C     TLEN1          = TOTAL LENGTH OF THE SLAB IN THE Y-DIRECTION
C                      OF THE GRID
C     TLEN2          = TOTAL LENGTH OF THE SLAB IN THE X-DIRECTION
C     TMP            = ICE MELTING TEMPERATURE.
C     TOFF           = OFF TIME OF STEP HEAT INPUT.
```

```
C       TOLD           = NON-DIMENSIONAL TEMPERATURE AT THE PREVIOUS
C                          ITERATION.
C       TON            = ON TIME OF STEP HEAT INPUT.
C       TREF           = REFERENCE TEMPERATURE.
C       TX1,TX2        = CONSTANT TEMPERATURE AT LOWER, UPPER, AND SIDE
C       TX3,TX4          BOUNDARIES OF SLAB.
C       VARICE         = SUBROUTINE USE TO FIND ICE SHAPE
C       WI             = INITIAL ACCELLERATION PARAMETER
C                        FOR TIME STEPS NIN
C       WM             = INTERMEDIATE ACELLERATION PARAMETER
C                        FOR TIME STEPS NMED
C       WF             = FINAL ACCELERATION PARAMETER
C      -----------------------------------------------------------------
C
      DIMENSION HEAD(53,80),ITYPE(99,99),IP(99,99),MM(29),NO(29),NOX(29)
      DIMENSION IPOLD(99,99),MX(99)
      DOUBLE PRECISION T(99,99),TO(99,99),DY(99),AAK(99,99),YP(99)
      DOUBLE PRECISION TOLD(99,99),H(99,99),HO(99,99),PA(99,99),X(99,99)
      DOUBLE PRECISION H2(99),QV(29),TE(99,99),CONST(99,99)
      DOUBLE PRECISION R(99,99),RR(99,99),S(99,99),SS(99,99),SO(99,99)
      DOUBLE PRECISION A(99,99),B(99,99),C(99,99),D(99,99),ZNOT(99,99)
      DOUBLE PRECISION ALP(29),EL(29),RX(99,99),AK(29),XP(99),DX(99)
      DOUBLE PRECISION ELX(29),AKX(29),ALPX(29),TLAG(29),RATE(99)
      DOUBLE PRECISION Q(29),HT(99,99),TON(29),TOFF(29),PHEAT(99)
      COMMON /SEA1/HSMP,HLMP,AKS,AKL,YL1,CPS,CPL,ZOTOT1,ZOTOT2,INC,DES
      COMMON /SEA2/IRF,NFSP,IRP,JSH,N,M,TIN,TREF,VA6,TMP,PSS,TR
      COMMON /SEA3/ZY,X2,X3,Z2,Z10,Z11,Z12,X1
      COMMON /SEA4/ALPHA,IS,IFIN,JJ,LLL,JS
      COMMON /SEA5/NP,L,JMAX,ISH,KSH,HAFP,NISP
      COMMON /SEA6/IBC1,IBC2,IBC3,IBC4,TG1,TG2,TG3,TG4
      DATA IN/1/,IO/6/,JO/56/
C      -----------------------------------------------------
C                            INPUT  DATA
C      -----------------------------------------------------
      DO 5 I=1,20
    5 READ(IN,700)(HEAD(I,J),J=1,72)
      READ(IN,701)L,NX
      DO 10 I=21,24
   10 READ(IN,700)(HEAD(I,J),J=1,72)
      DO 15 K=1,L
   15 READ(IN,702) NO(K),EL(K),AK(K),ALP(K)
      DO 20 I=25,27
   20 READ(IN,700)(HEAD(I,J),J=1,72)
      DO 25 K=1,NX
   25 READ(IN,702)NOX(K),ELX(K),AKX(K),ALPX(K)
      DO 30 I=28,30
   30 READ(IN,700)(HEAD(I,J),J=1,72)
      READ(IN,703)IH,L1,L2,IJ,IHQ
      DO 35 I=31,32
   35 READ(IN,700)(HEAD(I,J),J=1,72)
      DO 40 I=1,NX
   40 READ(IN,705)Q(I),TON(I),TOFF(I),QV(I),TLAG(I)
      DO 45 I=33,35
   45 READ(IN,700)(HEAD(I,J),J=1,72)
```

```
      READ(IN,706)IBC1,IBC2,IBC3,IBC4
      READ(IN,716)TX1,TX2,TX3,TX4
      READ(IN,707)TG1,H1,TG2,TG3,H3,TG4,H4,TIN,TREF
      MX(1)=1
      MX(2)=NOX(1)+1
      TLEN2=ELX(1)
      IF(NX.EQ.1)GO TO 50
      DO 50 I=3,NX+1
      TLEN2=TLEN2+ELX(I-1)
      MX(I)=MX(I-1)+NOX(I-1)
   50 CONTINUE
      N=MX(NX+1)
      READ(IN,739)(H2(I),I=2,N)
      DO 55 I=36,38
   55 READ(IN,700)(HEAD(I,J),J=1,72)
      READ(IN,709)IG,HLAM,ALPL,AKL,DEL,CPL,DES,CPS,TMP,NP
      DO 60 I=1,NP
   60 READ(IN,715)XP(I),YP(I),RATE(I)
      DO 65 I=39,41
   65 READ(IN,700)(HEAD(I,J),J=1,72)
      READ(IN,711)DTAUI,NISP,DTAUM,NMSP,DTAUF
      READ(IN,713)KENTH,ISH,IRF,NFSP,KSH,NHSP
      READ(IN,712)IFREQ,NTSP,JCOUNT,EPSI
      READ(IN,711)WI,NIN,WM,NMED,WF
      DO 70 I=42,51
   70 READ(IN,700)(HEAD(I,J),J=1,72)
      READ(IN,708)METHOD
      READ(IN,710)THETA
      DO 75 I=52,53
   75 READ(IN,700)(HEAD(I,J),J=1,72)
      READ(IN,710)ALPHA
C     --------------------------------------------------------
C                  INTERFACE  NODE  NUMBERING
C     --------------------------------------------------------
      MM(1)=1
      MM(2)=NO(1)+1
      TLEN1=EL(1)
      DO 80 I=3,L+1
      MM(I)=MM(I-1)+NO(I-1)-1
   80 TLEN1=TLEN1+EL(I-1)
      NODE=MM(L2)
      NO1=MM(IJ)
      NO2=MM(IJ+1)
      M=MM(L+1)
      MM(L+2)=M+1
C     --------------------------------------------------------
C                        PRINT THE DATA
C     --------------------------------------------------------
      DO 85 I=1,20
   85 WRITE(IO,775)(HEAD(I,J),J=1,72)
      MZ=M-1
      NZ=N-1
      WRITE(IO,717)L,NX,MZ,NZ,TLEN1,TLEN2
      DO 90 I=21,24
```

```
 90 WRITE(IO,775)(HEAD(I,J),J=1,72)
    DO 95 K=1,L
 95 WRITE(IO,702)NO(K),EL(K),AK(K),ALP(K)
    DO 100 I=25,27
100 WRITE(IO,775)(HEAD(I,J),J=1,72)
    DO 105 K=1,NX
105 WRITE(IO,702)NOX(K),ELX(K),AKX(K),ALPX(K)
    DO 110 I=28,30
110 WRITE(IO,775)(HEAD(I,J),J=1,72)
    IF(IH.EQ.2)WRITE(IO,720)NODE,L1,L2
    IF(IH.EQ.3)WRITE(IO,718)IJ,NO1,NO2
    IF(IH.EQ.1) GO TO 135
    IF(IHQ.NE.1) GO TO 120
    WRITE(IO,721)
    II=0
    DO 115 I=1,NX
    IF(Q(I).EQ.0.)GO TO 115
    II=II+1
    WRITE(IO,733)II,Q(I)
115 CONTINUE
    GO TO 140
120 WRITE(IO,725)
    II=0
    DO 130 I=1,NX
    IF(QV(I).EQ.0.)GO TO 130
    II=II+1
    WRITE(IO,722)II,QV(I),TON(I),TOFF(I),TLAG(I)
130 CONTINUE
    IF(IHQ.EQ.2)WRITE(IO,724)
    IF(IHQ.EQ.3)WRITE(IO,726)
    IF(IHQ.EQ.4)WRITE(IO,727)
    GO TO 140
135 WRITE(IO,719)
140 DO 145 I=33,35
145 WRITE(IO,775)(HEAD(I,J),J=1,72)
    IF(IBC1.EQ.2)WRITE(IO,730)TG1,H1
    IF(IBC1.EQ.1)WRITE(IO,729)TX1
    IF(IBC2.EQ.1)WRITE(IO,732)TX2
    IF(IBC3.EQ.2)WRITE(IO,742)TG3,H3
    IF(IBC3.EQ.1)WRITE(IO,741)TX3
    IF(IBC4.EQ.2)WRITE(IO,744)TG4,H4
    IF(IBC4.EQ.1)WRITE(IO,743)TX4
    WRITE(IO,736)TIN,TREF
    IF(IBC2.NE.2)GO TO 150
    WRITE(IO,734)TG2
    WRITE(IO,739)(H2(I),I=2,N)
150 WRITE(IO,775)(HEAD(33,J),J=1,72)
    IF(IG.EQ.1) GO TO 160
    WRITE(IO,738)
    WRITE(IO,775)(HEAD(36,J),J=1,72)
    WRITE(IO,775)(HEAD(35,J),J=1,72)
    WRITE(IO,775)(HEAD(37,J),J=1,72)
    WRITE(IO,775)(HEAD(38,J),J=1,72)
    WRITE(IO,740)HLAM,AKL,ALPL,DEL,CPL,DES,CPS,TMP,NP
```

```
      XP(NP)=TLEN2
      DO 155 I=1,NP
      IF(YP(I).GT.EL(L))YP(I)=EL(L)
  155 WRITE(IO,723)XP(I),YP(I),RATE(I)
      GO TO 165
  160 WRITE(IO,753)
      WRITE(IO,775)(HEAD(33,J),J=1,72)
  165 DO 170 I=39,41
  170 WRITE(IO,775)(HEAD(I,J),J=1,72)
      WRITE(IO,754)DTAUI,NISP,DTAUM,NMSP,DTAUF
      WRITE(IO,728)KENTH
      IF(ISH.NE.1) WRITE(IO,756)KSH
      IF(IRF.NE.1) WRITE(IO,758)NFSP,NHSP
      WRITE(IO,760)IFREQ,NTSP,JCOUNT,EPSI
      WRITE(IO,755)WI,NIN,WM,NMED,WF
      DO 175 I=42,51
  175 WRITE(IO,775)(HEAD(I,J),J=1,72)
      WRITE(IO,783)METHOD
      WRITE(IO,782)THETA
      DO 180 I=52,53
  180 WRITE(IO,775)(HEAD(I,J),J=1,72)
      WRITE(IO,784)ALPHA
      WRITE(IO,775)(HEAD(1,J),J=1,72)
      WRITE(IO,762)
      WRITE(IO,775)(HEAD(18,J),J=1,72)
      WRITE(IO,775)(HEAD(1,J),J=1,72)
C     -----------------------------------------------------
C                     INITIAL CONDITIONS
C     -----------------------------------------------------
      IRP=0
      NIP=0
      PSS=0.
      JSH=1
      DUMAX=0.
      ISKIP=0
      ITER=0
      JJ=2
      LLL=M
      IS=2
      IFIN=N
      JS=1
      TR=0.0001/TREF
      G2TIME=TG2/TREF
      G1TIME=TG1/TREF
      G3TIME=TG3/TREF
      G4TIME=TG4/TREF
      TMP=TMP/TREF
      X2TIME=TX2/TREF
      X1TIME=TX1/TREF
      X3TIME=TX3/TREF
      X4TIME=TX4/TREF
      DO 185 I=1,NP
      YP(I)=YP(I)/12.
  185 XP(I)=XP(I)/12.
```

```
      DO 190 J=1,L
  190 EL(J)=EL(J)/12.
      DO 195 J=1,NX
  195 ELX(J)=ELX(J)/12.
      TIN=TIN/TREF
      DO 200 I=1,NX
      TLAG(I)=TLAG(I)/3600.
      TON(I)=TON(I)/3600.
  200 TOFF(I)=TOFF(I)/3600.
      DTAUI=DTAUI/3600.
      DTAUM=DTAUM/3600.
      DTAUF=DTAUF/3600.
      TAU=0.
      DTAU=DTAUI
      LA=0
      INC=MM(L)
      AKS=AK(L)
      DO 205 K=1,L
      DO 205 J=MM(K)+1,MM(K+1)
  205 DY(J)=EL(K)/(NO(K)-1)
      MM(1)=2
      DO 210 K=1,NX
      DO 210 J=MX(K)+1,MX(K+1)
      DX(J)=ELX(K)/(NOX(K)+1)
      IF((K.EQ.1).OR.(K.EQ.NX))DX(J)=ELX(K)/NOX(K)
  210 IF((Q(K).NE.0).OR.(QV(K).NE.0))DX(J)=ELX(K)/(NOX(K)-1.)
      DY(1)=DY(2)
      DX(1)=DX(3)
      DY(M+1)=DY(M)
      DX(N+1)=DX(N-1)
      VA6=((TMP+TR)*TREF-(DEL*(CPS*(TMP+TR)*TREF/DES+HLAM)/CPL))/TREF
      YL1=DY(M)*DY(M)
      AK(L+1)=AK(L)
      ALP(L+1)=ALP(L)
      AKX(NX+1)=AKX(NX)
      ALPX(NX+1)=ALPX(NX)
      MX(NX+2)=N+1
      II=1
      K=1
      DO 220 J=1,M+1
      DO 220 I=1,N+1
      IF(I.EQ.(MX(II+1)+1))II=II+1
      IF(J.EQ.(MM(K+1)+1))K=K+1
      IF(I.LE.MX(2))II=1
      IF(J.LE.MM(2))K=1
      AAK(I,J)=AK(K)
      PA(I,J)=AK(K)/ALP(K)
      IF((K.NE.IJ).OR.(IH.NE.3)) GO TO 215
      AAK(I,J)=AKX(II)
      PA(I,J)=AKX(II)/ALPX(II)
  215 C(I,J)=AAK(I,J)/(2.*DX(I)*DX(I)*PA(I,J))
  220 D(I,J)=AAK(I,J)/(2.*DY(J)*DY(J)*PA(I,J))
C     --------------------------------------------------
C                    PRINT OUT HEATER NODES
```

```
C     ---------------------------------------------------
      KK=0
      DO 225 K=1,NX
      IBM=0
      DO 225 I=MX(K)+1,MX(K+1)
      II=I-1
      IF((Q(K).EQ.0.).OR.(IBM.EQ.1)) GO TO 225
      KK=KK+1
      IIE=MX(K+1)-1
      WRITE(IO,764)KK,II
      WRITE(IO,765)KK,IIE
      IBM=1
  225 CONTINUE
C     ---------------------------------------------------
C                         FIND ICE SHAPE
C     ---------------------------------------------------
      CALL TYPE(XP,YP,NP,ITYPE,JMAX,DY(M),DX,N,INC)
      IF(NP.LE.2)SM=ABS((YP(2)-YP(1))/(XP(2)-XP(1)))
      MM(L+1)=INC+JMAX
      M=MM(L+1)
      MM(L+2)=M+1
      IF(ISKIP.EQ.2)GO TO 230
      WRITE(IO,781)
      WRITE(IO,780)((ITYPE(I,J),I=2,N),J=INC,M)
C     ---------------------------------------------------
C        PRINT THE INITIAL TEMPERATURE PROFILE AND ENTHALPYS
C     ---------------------------------------------------
  230 DO 240 I=2,N
      DO 240 J=1,M
      T(I,J)=TIN
      TO(I,J)=TIN
      IF(ITYPE(I,J).NE.0)GO TO 235
      T(I,J)=0.0
      TO(I,J)=0.0
  235 H(I,J)=CPL*TREF*T(I,J)+DEL*(CPS*(TMP+TR)*TREF/DES+HLAM)-(TMP+TR)
     1*CPL*TREF
      IF(TIN.LE.TMP) H(I,J)=CPS*T(I,J)*TREF
      HO(I,J)=H(I,J)
      IPOLD(I,J)=1
      IF(TIN.GT.TMP)IPOLD(I,J)=2
      IF(TIN.GT.(TMP+TR))IPOLD(I,J)=3
  240 TE(I,J)=T(I,J)*TREF
      WRITE(IO,766) TAU
      DO 245 K=2,L+1
      KK=K-1
      KK2=K-2
      WRITE(IO,767) KK
      DO 245 J=MM(KK),MM(K)
  245 WRITE(IO,768)(TE(I,J),I=2,N)
      DO 250 J=INC,M
  250 IF(KENTH.EQ.2)WRITE(IO,769) (H(I,J),I=2,N)
      INC1=INC+1
      HSMP=CPS*TMP*TREF
      HLMP=DEL*(CPS*(TMP+TR)*TREF/DES+HLAM)
      HAFP=(HSMP+HLMP)/2.
C     -----------------------------------------------------
C                  START OF NEW TIME STEP
C     -----------------------------------------------------
  255 ICOUNT=0
      KCOUNT=0
      NIP=NIP+1
      MM(1)=1
C     -----------------------------------------------------
C                  DETERMINE THE TIME STEP
C     -----------------------------------------------------
      IF(NIP.GE.NISP+1) DTAU=DTAUM
      IF((JSH.NE.3).AND.(NIP.GE.NISP+NMSP+1)) DTAU=DTAUF
      LA=LA+1
      IF((METHOD.NE.1).OR.(THETA.GE.0.5))GO TO 270
      STA=0.
      DO 260 J=1,M+1
      DO 260 I=1,N+1
      BILITY=2.*(DX(I)**2+DY(J)**2)*AAK(I,J)/(PA(I,J)*(DX(I)*DY(J))**2)
  260 STA=AMAX1(STA,BILITY)
      DTAUO=1./STA
  265 IF(DTAU.LE.DTAUO)GO TO 270
      WRITE(IO,*)'THE TIME STEP INPUT WILL LEAD TO AN UNSTABLE SOLUTION'
      WRITE(IO,*)'INPUT NEW TIME STEP (MUST BE LESS THAN',DTAUO,'HRS)'
      READ(9,*)DTAU
      GO TO 265
  270 A1=1./DTAU
      W=1.
      IF((METHOD.NE.1).OR.(THETA.LT.0.5))GO TO 275
      W=WI
      IF(NIP.GE.NIN+1)W=WM
      NFIN=NMED+NIN
      IF(NIP.GE.NFIN+1)W=WF
  275 CALL ICE(NIP,T,H,TO,HO,MM,YP,RATE,DTAU,EL,XP,ITYPE,DY,DX,AAK,PA)
      M=MM(L+1)
      DY(M+1)=DY(M)
C     -----------------------------------------------------
C           SAVE TEMPERATURE FROM PREVIOUS TIME STEP
C     -----------------------------------------------------
      DO 280 I=2,N
      DO 280 J=2,M
      TO(I,J)=T(I,J)
  280 IF(J.GE.INC) HO(I,J)=H(I,J)
C     -----------------------------------------------------
C                  CALCULATION OF NEW TIME
C     -----------------------------------------------------
C
      TAU=TAU+DTAU
C
C     -----------------------------------------------------
C           CALCULATION OF TIME DEPENDANT CONSTANTS
C     -----------------------------------------------------
      VA3=DTAU*TREF/2.
      IHOP=0
  285 CALL SOLVE(THR,THL,THT,THB,ATR,ATL,ATT,ATB,N,M,THETA,METHOD,NIP,
     1IHOP)
C     ------------------------------------------------------------------
C                           HEAT TERM INCLUSION
C     ------------------------------------------------------------------
      DO 295 I=1,NX
      IF(IHQ.EQ.1)Q1=Q(I)
      IF(IHQ.EQ.2)Q1=QHEAT2(TAU,DTAU,TON(I),TOFF(I),QV(I),TLAG(I),THETA)
      IF(IHQ.EQ.3)Q1=QHEAT3(TAU,DTAU,TON(I),TOFF(I),QV(I),TLAG(I),THETA)
      IF(IHQ.EQ.4)Q1=QHEAT4(TAU,DTAU,TON(I),TOFF(I),QV(I),TLAG(I),THETA)
      Q2=2.*3.4121*144.*Q1
      DO 295 II=MX(I)+1,MX(I+1)
      DO 290 K=1,L
      DO 290 J=MM(K)+1,MM(K+1)
      HT(II,J)=Q2*DY(NO2)*DY(NO2)/(TREF*AK(IJ)*EL(IJ))
  290 IF((J.LT.NO1).OR.(J.GT.NO2))HT(II,J)=0.
  295 PHEAT(II)=2.*Q2*DY(MM(L1+1)+1)/(TREF*AK(L1))
      DO 300 J=1,M+1
      HT(1,J)=HT(2,J)
  300 HT(N+1,J)=HT(N,J)
      DO 305 I=1,N+1
      HT(I,1)=HT(I,2)
  305 HT(I,M+1)=HT(I,M)
      PHEAT(N+1)=PHEAT(N)
      DO 310 J=1,M+1
      DO 310 I=1,N+1
      A(I,J)=1./(A1+(C(I,J)*(THR+THL)+D(I,J)*(THT+THB)))
  310 B(I,J)=A1-(C(I,J)*(ATR+ATL)+D(I,J)*(ATT+ATB))
C     -------------------------------------------------------
C                   START OF NEW ITERATION
C     -------------------------------------------------------
  315 DO 355 K=1,L
      DO 355 J=MM(K)+1,MM(K+1)
      DO 355 II=1,NX
      DO 355 I=MX(II)+1,MX(II+1)
C     -------------------------------------------------------
C                         SET COUNTERS
C     -------------------------------------------------------
      Z1=1.
      Z2=1.
      Z3=1.
      Z5=1.
      Z6=1.
      Z7=1.
      Z8=0.
      Z9=0.
      Z16=1.
      Z17=1.
      IF(J.NE.2)Z1=0.
      IF(J.NE.MM(K+1))Z2=0.
      IF((J.EQ.2).OR.(J.EQ.M))Z2=0.
      IF((J.NE.NODE).OR.(IH.NE.2)) Z3=0.
      IF((IH.NE.3).OR.(J.NE.NO1)) Z5=0.
      IF((IH.NE.3).OR.(J.LE.NO1).OR.(J.GE.NO2)) Z6=0.
      IF((IH.NE.3).OR.(J.NE.NO2)) Z7=0.
      IF((I.EQ.MX(II)+1).AND.(HT(I,NO2).NE.0.))Z9=1.
      IF((I.EQ.MX(II+1)).AND.(HT(I,NO2).NE.0.))Z8=1.
      IF(I.NE.2)Z16=0.
      IF(I.NE.N)Z17=0.
      IF((Z8.EQ.1.).AND.(Z17.EQ.1))Z8=0.
      IF((Z9.EQ.1.).AND.(Z16.EQ.1))Z9=0.
      ZA2=0.
      ZA3=0.
      ZA4=0.
      ZA5=0.
      ZA6=0.
      ZA7=0.
      ZA8=0.
      ZA9=0.
      ZNOT(I,J)=0.
      IF(ITYPE(I,J).NE.0)GO TO 320
      R(I,J)=0.
      RR(I,J)=0.
      S(I,J)=0.
      SS(I,J)=0.
      SO(I,J)=0.
      X(I,J)=0.
      RX(I,J)=0.
      GO TO 355
  320 IF(ITYPE(I,J).EQ.2)ZA2=1.
      IF(ITYPE(I,J).EQ.3)ZA3=1.
      IF(ITYPE(I,J).EQ.4)ZA4=1.
      IF(ITYPE(I,J).EQ.5)ZA5=1.
      IF(ITYPE(I,J).EQ.6)ZA6=1.
      IF(ITYPE(I,J).EQ.7)ZA7=1.
      IF(ITYPE(I,J).EQ.8)ZA8=1.
      IF((ITYPE(I,J).EQ.9).AND.(ITYPE(I,J+1).EQ.0))ZA9=1.
      IF((ITYPE(I,J).NE.7).AND.(ITYPE(I,J).NE.8))GO TO 325
      Z16=0.
      Z17=0.
  325 ZY=ZA2+ZA5+ZA6+ZA8+ZA9
      ZX2=ZA3+ZA5+ZA7+ZA8
      ZX1=ZA4+ZA6+ZA7+ZA8
      ZX4=Z16+ZA3+ZA5
      ZX5=Z17+ZA4+ZA6
      XTOT1=1.-ZX1-Z17
      XTOT2=1.-ZX2-Z16
      Z58=Z5*(Z8+Z9)
      TOLD(I,J)=T(I,J)
      BOT=0.
      TOP=0.
      ALEFT=0.
      RIGHT=0.
      IF((J.GT.2).OR.(IBC1.EQ.2)) GO TO 330
C     -----------------------------------------------------
C          CONSTANT BOUNDARY CONDITION AT SUBSTRATE
C     -----------------------------------------------------
      T(I,2)=X1TIME
      TOLD(I,2)=X1TIME
      JJ=3
      IF(METHOD.NE.3)GO TO 355
      JJ=MM(L+1)
      LLL=3
      GO TO 355
  330 IF((METHOD.GE.4).AND.(IBC1.EQ.1).AND.(J.EQ.3))BOT=1.
C     -------------------------------------------------------
C           CONSTANT BOUNDARY CONDITION FOR AB. SHIELD
C     -------------------------------------------------------
      IF((IBC2.EQ.2).OR.(ITYPE(I,J).EQ.9).OR.(ZA7.EQ.1)) GO TO 335
      IF((ITYPE(I,J).EQ.1).OR.(ZA3.EQ.1).OR.(ZA4.EQ.1)) GO TO 335
      T(I,J)=X2TIME
      TOLD(I,J)=X2TIME
      ZNOT(I,J)=1.
      LLL=MM(L+1)-1
      IF(METHOD.NE.3)GO TO 355
      JJ=MM(L+1)-1
      LLL=3
      IF(IBC1.EQ.2)LLL=2
      GO TO 355
  335 IF((METHOD.GE.4).AND.(IBC2.EQ.1).AND.(J.EQ.M-1))TOP=1.
C     -------------------------------------------------------
C          CONSTANT BOUNDARY CONDITION FOR LEFT INTERFACE
C     -------------------------------------------------------
      IF((I.NE.2).OR.(IBC3.EQ.2)) GO TO 340
      T(2,J)=X3TIME
      TOLD(2,J)=X3TIME
      IS=3
      IF(METHOD.NE.3)GO TO 355
      IS=N
      IFIN=3
      GO TO 355
  340 IF((METHOD.GE.4).AND.(IBC3.EQ.1).AND.(I.EQ.3))ALEFT=1.
C     -------------------------------------------------------
C          CONSTANT BOUNDARY CONDITION FOR RIGHT INTERFACE
C     -------------------------------------------------------
      IF((I.NE.N).OR.(IBC4.EQ.2)) GO TO 345
      T(N,J)=X4TIME
      TOLD(N,J)=X4TIME
      IFIN=N-1
      IF(METHOD.NE.3)GO TO 355
      IS=N-1
      IFIN=2
      IF(IBC3.EQ.1)IFIN=3
      GO TO 355
  345 IF((METHOD.GE.4).AND.(IBC4.EQ.1).AND.(I.EQ.N-1))RIGHT=1.
C     ------------------------------------------------------------------
C          CALCULATION OF MATRIX COEFFICIENTS IN THE COMPOSITE BODY
C     ------------------------------------------------------------------
      T(1,J)=G3TIME
      TO(1,J)=G3TIME
      T(I,1)=G1TIME
      TO(I,1)=G1TIME
  355 CONTINUE
  360 DUMAX=0.
      IF(METHOD.GE.4)GO TO 375
C     -----------------------------------------------------------------
C            CALCULATING TEMPERATURES AND ENTHALPIES IN THE GRID
C     -----------------------------------------------------------------
      DO 370 J=JJ,LLL,JS
      DO 370 I=IS,IFIN,JS
      IF((ITYPE(I,J).EQ.0).OR.(ZNOT(I,J).EQ.1.))GO TO 370
      IF(METHOD.NE.2)GO TO 365
      THOP=.5*(I+J+NIP)
      JHOP=(I+J+NIP)/2
      IF((THOP.EQ.JHOP).AND.(IHOP.EQ.0))GO TO 370
      IF((THOP.NE.JHOP).AND.(IHOP.EQ.1))GO TO 370
  365 TOLD(I,J)=T(I,J)
      Z12=1.
      Z11=0.
      Z10=0.
      H(I,J)=R(I,J)*T(I+1,J)+RR(I,J)*T(I-1,J)+S(I,J)*T(I,J+1)+
     1SS(I,J)*T(I,J-1)+SO(I,J)+T(I,J)*RX(I,J)
      IF(X(I,J).NE.0.)CALL PHASE(IP,I,J,H,X,LAK,PA)
      T(I,J)=(H(I,J)+X(I,J))/(PA(I,J)*TREF)
  370 CONTINUE
C     ---------------------------------------------------
C                    USE DIFERENT METHODS
C     ---------------------------------------------------
C                     HOPSCOTCH AND ADE
C     ---------------------------------------------------
      IF((METHOD.NE.2).AND.(METHOD.NE.3))GO TO 375
      IF(IHOP.EQ.1)GO TO 405
      IHOP=1
      GO TO 285
C     ---------------------------------------------------
C                    ADI AND TIME-SPLITTING
C     ---------------------------------------------------
  375 IF((METHOD.NE.4).AND.(METHOD.NE.5))GO TO 380
      CALL TDMA(R,RR,S,SS,SO,T,NIP,PA,H,X,TOLD,ITYPE)
      GO TO 405
C     ---------------------------------------------------
C                 STRONGLY IMPLICT PROCEDURE
C     ---------------------------------------------------
C
  380 IF(METHOD.NE.6)GO TO 385
      CALL SIP(R,RR,S,SS,SO,T,H,PA,TREF,X,CPL,VA6,TOLD,ICOUNT,ITYPE)
C
C     ---------------------------------------------------
C             MODIFIED STRONGLY IMPLICT PROCEDURE
C     ---------------------------------------------------
C
  385 IF(METHOD.NE.7)GO TO 390
      CALL MSIP(R,RR,S,SS,SO,T,H,PA,TREF,X,CPL,VA6,TOLD,ITYPE)
C
C     ---------------------------------------------------
C            CHECK CONVERGENCE OF ITERATIVE METHODS
```

```
C     ------------------------------------------------------
  390 DO 400 J=JJ,LLL
      DO 400 I=IS,IFIN
      IF(ITYPE(I,J).EQ.0) GO TO 400
      IF(ABS(T(I,J)).LT..00001) GO TO 395
      DIF=100.*ABS((T(I,J)-TOLD(I,J))/T(I,J))
      DUMAX=AMAX1(DIF,DUMAX)
      GO TO 400
  395 DIF=100.*ABS(T(I,J)-TOLD(I,J))
      DUMAX=AMAX1(DIF,DUMAX)
  400 CONTINUE
      IF(THETA.GE.0.5)ITER=1
C     ------------------------------------------------------------------
C                 DETERMINATION OF WHETHER THE MAXIMUM
C                 NUMBER OF ITERATIONS HAS BEEN EXCEEDED
C     ------------------------------------------------------------------
C                 CHECK TO SEE IF NEED ANY MORE ITERATION
C     ------------------------------------------------------------------
  405 ICOUNT=ICOUNT+1
      IF((ITER.NE.0).AND.(ICOUNT.LT.JCOUNT).AND.(DUMAX.GT.EPSI))GOTO 360
      TFREQ=IFREQ/10
C     ------------------------------------------------------
C            CORRECTING METHOD OF ASSUMED STATES
C     ------------------------------------------------------
      IF((METHOD.EQ.2).OR.(METHOD.EQ.3).OR.(IG.EQ.1))GO TO 415
      IF(((METHOD.EQ.1).AND.(THETA.LT.0.5)).OR.(JSH.NE.1))GO TO 415
      DI=0.
      DIFMAX=0.
      DO 410 J=INC,LLL
      DO 410 I=IS,IFIN
      IF(ITYPE(I,J).EQ.0)GO TO 410
      IF((ITYPE(I,J).EQ.9).AND.(ITYPE(I,J+1).EQ.0))GO TO 410
      IF((H(I,J).LE.HSMP).AND.(HO(I,J).GE.HLMP))H(I,J)=HLMP-1.
      IF((H(I,J).GE.HLMP).AND.(HO(I,J).LE.HSMP))H(I,J)=HSMP+1.
      IF(H(I,J).LE.HSMP)IP(I,J)=1
      IF((H(I,J).GT.HSMP).AND.(H(I,J).LT.HLMP))IP(I,J)=2
      IF(H(I,J).GE.HLMP)IP(I,J)=3
      IF(IPOLD(I,J).EQ.IP(I,J))GO TO 410
      DI=1.
      DIFMAX=AMAX1(DI,DIFMAX)
  410 CONTINUE
      KCOUNT=KCOUNT+1
      IF((KCOUNT.LT.10).AND.(DIFMAX.GT.0.5))GO TO 315
      KCOUNT=0
  415 IF(METHOD.NE.3)GO TO 425
      DO 420 J=JJ,LLL,JS
      DO 420 I=IS,IFIN,JS
  420 T(I,J)=(T(I,J)+TOLD(I,J))/2.
C     ------------------------------------------------------
C                    PRINT OUT OF PROGRAM
C     ------------------------------------------------------
  425 TITAU=TAU*3600.
      WRITE(IO,766)TITAU
      WRITE(IO,770)ICOUNT
      IF(LA.LT.IFREQ) GO TO 255
      LA=0
      DO 430 I=2,N
      DO 430 J=1,M
  430 TE(I,J)=T(I,J)*TREF
      MM(1)=2
      DO 435 K=2,L+1
      KK=K-1
      KKK=K-2
      WRITE(IO,767) KK
      DO 435 J=MM(KK),MM(K)
  435 WRITE(IO,768) (TE(I,J),I=2,N)
      IF(KENTH.EQ.1) GO TO 445
      DO 440 J=INC,M
  440 WRITE(IO,769)(H(I,J),I=2,N)
C     ---------------------------------------------------------------
C              DETERMINATION OF WHETHER THE PROGRAM SHOULD
C              BE TERMINATED BY NUMBER OF TIME STEPS
C     ---------------------------------------------------------------
  445 IF(NIP.LT.NTSP) GO TO 255
      IF(ISKIP.EQ.2)GO TO 999
      WRITE(IO,781)
      WRITE(IO,780)((ITYPE(I,J),I=2,N),J=INC,M)
C     ---------------------------------------------------------------
C                 FORMAT STATEMENTS FOR INPUT AND OUTPUT
C     ---------------------------------------------------------------
  700 FORMAT(72A1)
  701 FORMAT(56X,I3/56X,I3)
  702 FORMAT(7X,I2,7X,F10.5,9X,F13.6,8X,F13.6)
  703 FORMAT(53X,I3/53X,I3/53X,I3/53X,I3/53X,I3)
  705 FORMAT(2X,F10.4,5X,F6.2,5X,F6.2,7X,F10.4,8X,F6.2)
  706 FORMAT(53X,I3/53X,I3/53X,I3/53X,I3)
  707 FORMAT(46X,F13.6/46X,F13.6/46X,F13.6/46X,F13.6/46X,F13.6
     1/46X,F13.6/46X,F13.6/46X,F13.6/46X,F13.6//)
  708 FORMAT(50X,I2)
  709 FORMAT(61X,I3/43X,F13.6/43X,F13.6/43X,F13.6/43X,F13.6/43X,F13.6
     1/43X,F13.6/43X,F13.6/43X,F13.6/50X,I3)
  710 FORMAT(50X,F6.4)
  711 FORMAT(50X,F13.6/50X,I4/50X,F13.6/50X,I4/50X,F13.6)
  712 FORMAT(50X,I5/50X,I6/50X,I4/50X,F8.6)
  713 FORMAT(/50X,I3/68X,I2/68X,I2/58X,I3/50X,I3/50X,I3/)
  714 FORMAT(/' # OF NODES   LENGTH IN X-DIR    CONDUCTIVITY   DIFFUSIVI
     1TY')
  715 FORMAT(10X,F10.5,10X,F10.5,10X,F10.5)
  716 FORMAT(46X,F13.6/46X,F13.6/46X,F13.6/46X,F13.6/)
  717 FORMAT(/ ' TOTAL NUMBER OF SLABS',38X,'L=',I3/ ' NUMBER OF LAYERS
     1IN HEATER',32X,'NX=',I3/' TOTAL NUMBER OF NODES IN THE Y-DIRECTION
     2',19X,'M=',I3/' TOTAL NUMBER OF NODES IN THE X-DIRECTION',19X,'N='
     3,I3/' TOTAL LENGTH OF COMPOSITE SLAB IN THE Y-DIRECTION',6X,'TLEN1
     4=',F13.6,'INCHES'/' TOTAL LENGTH OF THE GRID IN THE X-DIRECTION',
     512X,'TLEN2=',F13.6,'INCHES')
  718 FORMAT(' INTERNAL HEAT GENERATION IN SLAB NUMBER',9X,' IJ=',
     1I3/33X,'BETWEEN NODE    NO1=',I3/36X,'AND NODE     NO2=',I3)
  719 FORMAT(//10X,'THERE  IS  NO  HEAT  SOURCE  PRESENT  ')
```

```
720 FORMAT(/ ' POINT HEAT SOURCE IS PRESENT AT',17X,'NODE=',I3/
   133X,'BETWEEN SLAB     L1=',I3/36X,'AND  SLAB     L2=',I3)
721 FORMAT(/ ' # HEATER       CONSTANT WATTAGE HEATER IN W/IN**2')
722 FORMAT(/5X,I2,5X,F13.5,5X,F13.5,3X,F13.5,3X,F13.5)
723 FORMAT(7X,'X-COORDINATE XP =',F10.5,7X,'Y-COORDINATE YP =',F10.5
   1,7X,'RATE OF ICE GROWTH',F10.5)
724 FORMAT(/ ' STEP FUNCTION HEAT INPUT IN WATTS/IN*IN')
725 FORMAT(/ ' HEATER #', 5X,'WATTS/IN*IN',7X,'TIME ON(SEC)',5X,
   1'TIME OFF(SEC)',5X,'TIME LAG(SEC)')
726 FORMAT(/ ' RAMP FUNCTION HEAT INPUT IN WATTS/IN*IN')
727 FORMAT(/ ' SINE FUNCTION HEAT INPUT IN WATTS/IN*IN')
728 FORMAT(/ ' PRINT ENTHALPYS IN ICE LAYER; KENTH=2',16X,'KENTH=',I3)
729 FORMAT(// ' CONSTANT TEMPERATURE   AT  J=1',17X,' TX1=',F13.6,
   1' DEG.F')
730 FORMAT(// ' CONVECTION OCCURS      AT  J=1'/11X,'AMBIENT TEMPERAT
   1URE',5X,'TG1=',F13.6,'DEG.F'/11X,'HEAT TRANSFER COEFF.',5X,'H1 =',
   2F13.6,'B.T.U/HR.FT.FT.DEG.F')
731 FORMAT(53X,I3)
732 FORMAT(// ' CONSTANT TEMPERATURE   AT  J=M',17X,'TX2=',F13.6,
   1'DEG.F')
733 FORMAT(/5X,I2,12X,F13.6)
734 FORMAT(// ' CONVECTION OCCURS      AT  J=M'/11X,'AMBIENT TEMPERAT
   1URE',5X,'TG2=',F13.6,'DEG.F'/11X,'HEAT TRANSFER COEFFICIENT  IN
   2B.T.U/HR.FT.FT.DEG.F'/)
736 FORMAT(/ ' THE INITIAL TEMPERATURE IN THE COMPOSITE SLAB  TIN =',
   1F13.6,'DEG.F'// ' THE REFERENCE TEMPERATURE',20X,' TREF =',F13.6,
   2'DEG.F')
738 FORMAT(// ' THE PHASE CHANGE IN THE ICE LAYER IS CONSIDERED ')
739 FORMAT(4X,F10.3,4X,F10.3,4X,F10.3,4X,F10.3,4X,F10.3)
740 FORMAT(/ ' LATENT HEAT OF ICE ',21X,' HLAM =',F13.6,' B.T.U./LB'/
   1' THERMAL CONDUCTIVITY OF WATER',12X,'AKL  =',F13.6,' B.T.U./HR.FT
   2.DEG.F'/ ' THERMAL DIFFUSIVITY OF WATER ',12X,'ALPL =',F13.6,
   3' FT.FT/HR.'/ ' DENSITY OF WATER ',24X,'DEL  =',F13.6,' LB/CU.FT'/
   4 ' SPECIFIC HEAT * DENSITY OF WATER ',8X,'CPL  =',F13.6,' B.T.U./
   5CU.FT.DEG.F'/ ' DENSITY OF ICE ',26X,'DES  =',F13.6,' LB/CU.FT.'
   6/ ' SPECIFIC HEAT * DENSITY OF ICE ',10X,'CPS  =',F13.6,' B.T.U./
   7CU.FT.F'/ ' ICE MELTING TEMPERATURE',18X,'TMP  =',F13.6,' DEG.F'
   8/7X,'NUMBER OF POINTS IN ICE SHAPE APPROXIMATION NP =',I3)
741 FORMAT(// ' CONSTANT TEMPERATURE   AT  I=1',17X,' TX3=',F13.6,
   1' DEG.F')
742 FORMAT(// ' CONVECTION OCCURS      AT  I=1'/11X,'AMBIENT TEMPERAT
   1URE',5X,'TG3=',F13.6,'DEG.F'/11X,'HEAT TRANSFER COEFF.',5X,'H3 =',
   2F13.6,'B.T.U/HR.FT.FT.DEG.F')
743 FORMAT(// ' CONSTANT TEMPERATURE   AT  I=N',17X,'TX4=',F13.6,
   1'DEG.F')
744 FORMAT(// ' CONVECTION OCCURS      AT  I=N'/11X,'AMBIENT TEMPERAT
   1URE',5X,'TG4=',F13.6,'DEG.F'/11X,'HEAT TRANSFER COEFF.',5X,'H4 =',
   2F13.6,'B.T.U/HR.FT.FT.DEG.F')
753 FORMAT(// ' THE PHASE CHANGE IN THE ICE LAYER IS NOT CONSIDERED')
754 FORMAT(/ ' INITIAL TIME STEP',26X,'DTAUI =',F13.6,'SECS'/28X,
   1'FOR TIME STEPS NISP   =',I3/ ' INTERMIDIATE TIME STEP',21X,
   2'DTAUM =',F13.6,'SECS'/28X,' FOR TIME STEPS NMSP  =',I3/
   3' FINAL TIME STEP',28X,'DTAUF =',F13.6,'SECS')
755 FORMAT(/ ' INITIAL ACCELLERATION PARAMETER',12X,'WI=',F5.2/20X,
     1'FOR TIME STEPS NIN =',I3/ ' INTERMEDIATE PARAMETER',20X,'WM =',
     2F5.2/20X,'FOR TIME STEPS NMED =',I3/' FINAL PARAMETER',30X,'WF =',
     3F5.2)
  756 FORMAT(/// ' SHEDDING ICE LAYER IS CONSIDERED AT NODE ',I2)
  758 FORMAT(/ ' INITIAL TEMPERATURE IS CHANGED FOR TIME STEPS NFSP =',
     1I3/ ' FOR SHEDDING AT 2ND CYCLE FOR TIME STEPS  NHSP=',I3)
  760 FORMAT(' FREQUENCY OF TIME STEP/PRINT OF OUTPUT',6X,
     1'IFREQ=',I5/ ' FOR TIME STEPS STOP THE PROGRAM',13X,'NTSP =',I6
     2/ ' MAXIMUM ITERATIONS PER TIME STEP  JCOUNT = ',I4
     3/ ' TEMPERATURES CONVERGE FOR 100*[T-TOLD]/T .LT.  ',F8.6)
  762 FORMAT(///28X,' TEMPERATURE PROFILE  IN  DEGREES F  ')
  764 FORMAT(/2X,'HEATER',I3, ' STARTS AT NODE IN X-DIRECTION',5X,
     1'NODEG = ',I3)
  765 FORMAT(/2X,'HEATER',I3, ' ENDS AT NODE IN X-DIRECTION',5X,
     1'NODG = ',I3)
  766 FORMAT(///35X,' TIME  TAU =',F15.6,' SECS')
  767 FORMAT(/5X,'LAYER ',I2)
  768 FORMAT(/5X,12F7.2)
  769 FORMAT(/5X,12F10.2)
  770 FORMAT(/2X,'PASS=',I3)
  775 FORMAT(' ',80A1)
  780 FORMAT(' ',12I2)
  781 FORMAT(// '  ELEMENT TYPE MATRIX FOR THE ICE LAYER;'//)
  782 FORMAT(/' THETA FOR COMBINED METHOD --    0= EXPLICIT; 1/2= CRANK
     1-NICOLSON; 1= IMPLICIT.'/10X,' VALUE OF  THETA = ',F5.3)
  783 FORMAT(/' VALUE FOR METHOD     = ',I3)
  784 FORMAT(/' VALUE FOR ALPHA ( 0.3 < ALPHA < 0.6 )    = ',F5.3)
  999 STOP
      END
C     ---------------------------------------------------------------------
C     ---------------------------------------------------------------------
C             SUBROUTINE FOR CONSTANT STEP FUNCTION HEAT SOURCE
C     ---------------------------------------------------------------------
C     ---------------------------------------------------------------------
      FUNCTION QHEAT2(TAU,DTAU,TON,TOFF,QV,TLAG,THETA)
      IF(TAU.GE.TLAG) GO TO 10
      QHEAT2=0.
      GO TO 20
   10 TAUR=TAU-TLAG-DTAU*(1.-THETA)
      IN=IFIX(TAUR/(TON+TOFF))
      IP=IN+1
      B=IN*TOFF+IP*TON
      C=IP*(TON+TOFF)
      QHEAT2=QV
      IF((B.LT.TAUR).AND.(TAUR.LT.C))QHEAT2=0.
   20 RETURN
      END
C     ---------------------------------------------------------------------
C     ---------------------------------------------------------------------
C                  SUBROUTINE FOR RAMP FUNCTION HEAT SOURCE
C     ---------------------------------------------------------------------
C     ---------------------------------------------------------------------
      FUNCTION QHEAT3(TAU,DTAU,TON,TOFF,QV,TLAG,THETA)
      IF(TAU.GE.TLAG) GO TO 10
      QHEAT3=0.
      GO TO 20
   10 TAUR=TAU-TLAG+DTAU*(THETA-1.)
      IN=IFIX(TAUR/(TON+TOFF))
      YP=TAUR-IN*(TON+TOFF)
      A=QV/TON
      QHEAT3=YP*A
      IP=IN+1
      B =IN*TOFF+IP*TON
      C =IP*(TON+TOFF)
      IF((B.LT.TAUR).AND.(TAUR.LT.C))QHEAT3=0.
   20 RETURN
      END
C     -----------------------------------------------------------------
C     -----------------------------------------------------------------
C               SUBROUTINE FOR SINE FUNCTION HEAT SOURCE
C     -----------------------------------------------------------------
C     -----------------------------------------------------------------
      FUNCTION QHEAT4(TAU,DTAU,TON,TOFF,QV,TLAG,THETA)
      IF(TAU.GE.TLAG) GO TO 10
      QHEAT4=0.
      GO TO 20
   10 TAUR=TAU-TLAG+DTAU*(THETA-1.)
      IN=IFIX(TAUR/(TON+TOFF))
      YP=TAUR-IN*(TON+TOFF)
      XP=YP*3.14159/TON
      QHEAT4=QV*SIN(XP)
      IP=IN+1
      B =IN*TOFF+IP*TON
      C =IP*(TON+TOFF)
      IF((B.LT.TAUR).AND.(TAUR.LT.C))QHEAT4=0.
   20 RETURN
      END
C     -----------------------------------------------------------------
C     -----------------------------------------------------------------
C     SUBROUTINE FOR CORRECTING THERMAL CONDUCTIVITIES FOR PHASE CHANGE
C     -----------------------------------------------------------------
C     -----------------------------------------------------------------
      SUBROUTINE CHANGE(H,IP,AAK,PA,VK1,VK2,VK3,VK4,THR,THL,THT,THB,I,J,
     1AK1,AK2,AK3,AK4,X,ITYPE)
      DOUBLE PRECISION H(99,99),X(99,99),PA(99,99),IP(99,99),AAK(99,99)
      DIMENSION ITYPE(99,99)
      COMMON /SEA1/HSMP,HLMP,AKS,AKL,YL1,CPS,CPL,XTOT1,XTOT2,INC,DES
      COMMON /SEA2/IRF,NFSP,IRP,JSH,N,M,TIN,TREF,VA6,TMP,PSS,TR
      COMMON /SEA3/ZY1,X2,X3,Z2,Z10,Z11,Z12,X1
      L=J
      DO 40 K=I-1,I+1
      IF(H(K,L).LE.HSMP) GO TO 10
      IF((H(K,L).GT.HSMP).AND.(H(K,L).LT.HLMP)) GO TO 20
      IF(H(K,L).GE.HLMP) GO TO 30
C     -----------------------------------------------------
C                    VALUES FOR SOLID ICE
C     -----------------------------------------------------
   10 IP(K,L)=1
```

```
      IF((J.EQ.INC).AND.(ITYPE(K,L+1).EQ.0)) GO TO 40
      AAK(K,L)=AKS
      PA(K,L)=CPS
      X(K,L)=0.
      GO TO 40
C     ---------------------------------------------------------
C                      VALUES FOR MELTING ICE
C     ---------------------------------------------------------
   20 IP(K,L)=2
      IF((J.EQ.INC).AND.(ITYPE(K,L+1).EQ.0)) GO TO 40
      V=(H(K,L)-HSMP)/(HLMP-HSMP)
      AAK(K,L)=(1.-V)*AKS+V*AKL
      PA(K,L)=(HLMP-HSMP)/(TR*TREF)
      X(K,L)=PA(K,L)*TMP*TREF-HSMP
      GO TO 40
C     ---------------------------------------------------------
C                         VALUES FOR WATER
C     ---------------------------------------------------------
   30 IP(K,L)=3
      IF((J.EQ.INC).AND.(ITYPE(K,L+1).EQ.0)) GO TO 40
      AAK(K,L)=AKL
      PA(K,L)=CPL
      X(K,L)=CPL*VA6*TREF
   40 CONTINUE
C     ---------------------------------------------------------
      K=I
      DO 80 L=J-1,J+1
      IF(H(K,L).LE.HSMP) GO TO 50
      IF((H(K,L).GT.HSMP).AND.(H(K,L).LT.HLMP)) GO TO 60
      IF(H(K,L).GE.HLMP) GO TO 70
C     ---------------------------------------------------------
C                        VALUES FOR SOLID ICE
C     ---------------------------------------------------------
   50 IP(K,L)=1
      IF(L.EQ.(INC-1))GO TO 80
      AAK(K,L)=AKS
      PA(K,L)=CPS
      X(K,L)=0.
      GO TO 80
C     ---------------------------------------------------------
C                       VALUES FOR MELTING ICE
C     ---------------------------------------------------------
   60 IP(K,L)=2
      IF(L.EQ.(INC-1))GO TO 80
      V=(H(K,L)-HSMP)/(HLMP-HSMP)
      AAK(K,L)=(1.-V)*AKS+V*AKL
      PA(K,L)=(HLMP-HSMP)/(TR*TREF)
      X(K,L)=PA(K,L)*TMP*TREF-HSMP
      GO TO 80
C     ---------------------------------------------------------
C                          VALUES FOR WATER
C     ---------------------------------------------------------
   70 IP(K,L)=3
      IF(L.EQ.(INC-1))GO TO 80
      AAK(K,L)=AKL
      PA(K,L)=CPL
      X(K,L)=CPL*VA6*TREF
   80 CONTINUE
C     ---------------------------------------------------
C            EVALUATING TERMS FOR THE POINTS IN QUESTION
C     ---------------------------------------------------
      CALL PHASE(IP,I,J,H,X,AAK,PA)
      AK1=(AAK(I,J)+AAK(I-1,J))/2.
      IF(I.EQ.2)AK1=AAK(I,J)
      AK2=(AAK(I,J)+AAK(I+1,J))/2.
      IF(I.EQ.N)AK2=AAK(I,J)
      AK3=(AAK(I,J)+AAK(I,J+1))/2.
      IF(J.EQ.M)AK3=AAK(I,J)
      AK4=(AAK(I,J)+AAK(I,J-1))/2.
      VK1=AK1*X1*XTOT2*THL
      VK2=AK2*X1*XTOT1*THR
      VK3=(1.-ZY1+Z2)*AK3*THT/YL1
      VK4=(1.+ZY1-Z2)*AK4*THB/YL1
      RETURN
      END
C     ----------------------------------------------------------------
C     ----------------------------------------------------------------
C             SUBROUTIINE TO DETERMINE CONSTANTS FOR PHASE CHANGE
C     ----------------------------------------------------------------
C     ----------------------------------------------------------------
      SUBROUTINE PHASE(IP,I,J,H,X,AAK,PA)
      DOUBLE PRECISION H(99,99),X(99,99),PA(99,99),AAK(99,99)
      DIMENSION IP(99,99)
      COMMON /SEA1/HSMP,HLMP,AKS,AKL,YL1,CPS,CPL,XTOT1,XTOT2,INC,DES
      COMMON /SEA2/IRF,NFSP,IRP,JSH,N,M,TIN,TREF,VA6,TMP,PSS,TR
      COMMON /SEA3/ZY1,X2,X3,Z2,Z10,Z11,Z12,X1
      Z10=0.
      Z11=0.
      Z12=0.
      IF(IP(I,J).EQ.1)GO TO 10
      IF(IP(I,J).EQ.2)GO TO 20
      IF(IP(I,J).EQ.3)GO TO 30
C     ---------------------------------------------------
C                        VALUES FOR SOLID ICE
C     ---------------------------------------------------
   10 AAK(I,J)=AKS
      PA(I,J)=CPS
      X(I,J)=0.
      Z12=1.
      GO TO 40
C     ---------------------------------------------------
C                       VALUES FOR MELTING ICE
C     ---------------------------------------------------
   20 V=(H(I,J)-HSMP)/(HLMP-HSMP)
      AAK(I,J)=(1.-V)*AKS+V*AKL
      PA(I,J)=(HLMP-HSMP)/(TR*TREF)
      X(I,J)=PA(I,J)*TMP*TREF-HSMP
      Z10=1.
      GO TO 40
C     ------------------------------------------------------
C                      VALUES FOR WATER
C     ------------------------------------------------------
   30 AAK(I,J)=AKL
      PA(I,J)=CPL
      X(I,J)=CPL*VA6*TREF
      Z11=1.
   40 RETURN
      END
C     ------------------------------------------------------
C     ------------------------------------------------------
C           SUBROUTINE FOR DETERMINING THE ICE SHAPE
C     ------------------------------------------------------
C     ------------------------------------------------------
      SUBROUTINE TYPE(XP,YP,NP,ITYPE,JMAX,DY,X1,M,INC)
      DOUBLE PRECISION XP(99),YP(99),X1(99)
      DIMENSION ITYPE(99,99)
      ISTOP=99
      DO 20 J=INC,ISTOP
      Y=(J-INC)*DY
      X=0.
      DO 20 I=2,M
      ITYPE(I,J)=0
      DO 10 K=1,NP-1
      IF((X.LT.XP(K)).OR.(X.GE.XP(K+1)))GO TO 10
      XS=XP(K)
      YS=YP(K)
      SM=(YP(K+1)-YP(K))/(XP(K+1)-XP(K))
   10 CONTINUE
      YLINE=SM*(X-XS)+YS
      IF(I.EQ.M)YLINE=YP(NP)
      IF(Y.LE.YLINE)ITYPE(I,J)=1
   20 X=X+X1(I)
C     ---------------------------------------------------------------
C                DETERMINING ICE SHAPE ON LEFT SIDE OF GRID
C     ---------------------------------------------------------------
      DO 30 J=INC+1,ISTOP
      IF(ITYPE(2,J).EQ.0)GO TO 30
      IF((ITYPE(2,J+1).NE.0).AND.(ITYPE(3,J).NE.0))ITYPE(2,J)=1
      IF((ITYPE(2,J+1).EQ.0).AND.(ITYPE(3,J).NE.0))ITYPE(2,J)=2
      IF((ITYPE(2,J+1).NE.0).AND.(ITYPE(3,J).EQ.0))ITYPE(2,J)=7
      IF((ITYPE(2,J+1).EQ.0).AND.(ITYPE(3,J).EQ.0))ITYPE(2,J)=8
   30 CONTINUE
C     ---------------------------------------------------------------
C            DETERMINING ICE SHAPE IN THE MIDDLE OF THE GRID
C     ---------------------------------------------------------------
      DO 4C J=INC+1,ISTOP
      DO 40 I=3,M-1
      IF(ITYPE(I,J).EQ.0)GO TO 40
      IF((ITYPE(I-1,J).NE.0).AND.(ITYPE(I+1,J).NE.0).AND.(ITYPE(I,J+1)
     1.NE.0))ITYPE(I,J)=1
      IF((ITYPE(I-1,J).NE.0).AND.(ITYPE(I+1,J).NE.0).AND.(ITYPE(I,J+1)
     1.EQ.0))ITYPE(I,J)=2
```

```
      IF((ITYPE(I-1,J).EQ.0).AND.(ITYPE(I+1,J).NE.0).AND.(ITYPE(I,J+1)
     1.NE.0))ITYPE(I,J)=3
      IF((ITYPE(I-1,J).NE.0).AND.(ITYPE(I+1,J).EQ.0).AND.(ITYPE(I,J+1)
     1.NE.0))ITYPE(I,J)=4
      IF((ITYPE(I-1,J).EQ.0).AND.(ITYPE(I+1,J).NE.0).AND.(ITYPE(I,J+1)
     1.EQ.0))ITYPE(I,J)=5
      IF((ITYPE(I-1,J).NE.0).AND.(ITYPE(I+1,J).EQ.0).AND.(ITYPE(I,J+1)
     1.EQ.0))ITYPE(I,J)=6
      IF((ITYPE(I-1,J).EQ.0).AND.(ITYPE(I+1,J).EQ.0).AND.(ITYPE(I,J+1)
     1.EQ.0))ITYPE(I,J)=8
      IF((ITYPE(I-1,J).EQ.0).AND.(ITYPE(I+1,J).EQ.0).AND.(ITYPE(I,J+1)
     1.NE.0))ITYPE(I,J)=7
   40 CONTINUE
C     -----------------------------------------------------------------
C               DETERMINING ICE SHAPE ON RIGHT SIDE OF GRID
C     -----------------------------------------------------------------
      DO 50 J=INC+1,ISTOP
      IF(ITYPE(M,J).EQ.0)GO TO 50
      IF((ITYPE(M,J+1).NE.0).AND.(ITYPE(M-1,J).NE.0))ITYPE(M,J)=1
      IF((ITYPE(M,J+1).EQ.0).AND.(ITYPE(M-1,J).EQ.0))ITYPE(M,J)=8
      IF((ITYPE(M,J+1).NE.0).AND.(ITYPE(M-1,J).EQ.0))ITYPE(M,J)=7
      IF((ITYPE(M,J+1).EQ.0).AND.(ITYPE(M-1,J).NE.0))ITYPE(M,J)=2
   50 CONTINUE
C     -----------------------------------------------------------------
C              SETTING VALUES OF ITYPE FOR THE COMPOSITE LAYER
C     -----------------------------------------------------------------
      DO 70 I=2,M
      DO 60 J=1,INC-1
   60 ITYPE(I,J)=1
   70 ITYPE(I,INC)=9
      J=INC
   80 ISUM=0
      DO 90 I=2,M
   90 ISUM=ISUM+ITYPE(I,J)
      IF(ISUM.EQ.0)GO TO 100
      J=J+1
      IF(J.GE.ISTOP)GO TO 100
      GO TO 80
  100 JMAX=J-1-INC
      RETURN
      END
C     -----------------------------------------------------------------
C     -----------------------------------------------------------------
C           SUBROUTINE SOLVE DETERMINES THE NUMERICAL METHOD USED
C     -----------------------------------------------------------------
C     -----------------------------------------------------------------
      SUBROUTINE SOLVE(THR,THL,THT,THB,ATR,ATL,ATT,ATB,N,M
     1,THETA,METHOD,NIP,IHOP)
      COMMON /SEA4/ALPHA,IS,IFIN,JJ,LLL,JS
      GO TO(10,20,40,60,80,100,110)METHOD
   10 THR=2.*THETA
      THL=2.*THETA
      THT=2.*THETA
      THB=2.*THETA
      ATR=2.*(1.-THETA)
      ATL=2.*(1.-THETA)
      ATT=2.*(1.-THETA)
      ATB=2.*(1.-THETA)
      GO TO 120
   20 IF(IHOP.EQ.1)GO TO 30
      THR=0.
      THL=0.
      THT=0.
      THB=0.
      ATR=2.
      ATL=2.
      ATT=2.
      ATB=2.
      THETA=0.
      GO TO 120
   30 THR=2.
      THL=2.
      THT=2.
      THB=2.
      ATR=0.
      ATL=0.
      ATT=0.
      ATB=0.
      THETA=1.
      GO TO 120
   40 IF(IHOP.EQ.1) GO TO 50
      THL=2.
      THB=2.
      ATR=2.
      ATT=2.
      THR=0.
      THT=0.
      ATL=0.
      ATB=0.
      IS=2
      IFIN=N
      JJ=2
      LLL=M
      JS=1
      THETA=.5
      GO TO 120
   50 THL=0.
      THB=0.
      ATR=0.
      ATT=0.
      THR=2.
      THT=2.
      ATL=2.
      ATB=2.
      IS=N
      IFIN=2
      JJ=M
      LLL=2
      JS=-1
      THETA=.5
      GO TO 120
   60 IF((.5*NIP).NE.(NIP/2))GO TO 70
      THL=0.
      THR=0.
      THT=2.
      THB=2.
      ATT=0.
      ATB=0.
      ATR=2.
      ATL=2.
      THETA=.5
      GO TO 120
   70 THL=2.
      THR=2.
      THT=0.
      THB=0.
      ATT=2.
      ATB=2.
      ATR=0.
      ATL=0.
      THETA=.5
      GO TO 120
   80 IF((.5*NIP).NE.(NIP/2))GO TO 90
      THR=0.
      THL=0.
      ATR=0.
      ATL=0.
      THT=4.*THETA
      THB=4.*THETA
      ATT=4.*(1.-THETA)
      ATB=4.*(1.-THETA)
      GO TO 120
   90 THT=0.
      THB=0.
      ATT=0.
      ATB=0.
      THR=4.*THETA
      THL=4.*THETA
      ATR=4.*(1.-THETA)
      ATL=4.*(1.-THETA)
      GO TO 120
  100 THR=2.*THETA
      THL=2.*THETA
      THT=2.*THETA
      THB=2.*THETA
      ATR=2.*(1.-THETA)
      ATL=2.*(1.-THETA)
      ATT=2.*(1.-THETA)
      ATB=2.*(1.-THETA)
      GO TO 120
  110 THR=2.*THETA
      THL=2.*THETA
      THT=2.*THETA
      THB=2.*THETA
      ATR=2.*(1.-THETA)
      ATL=2.*(1.-THETA)
      ATT=2.*(1.-THETA)
      ATB=2.*(1.-THETA)
  120 RETURN
      END
C     ----------------------------------------------------------------------
C     ----------------------------------------------------------------------
C            SUBROUTINE MSIP SOLVES AN N*M MATRIX OF EQUATIONS
C            USING THE MODIFIED STRONGLY IMPLICIT PROCEDURE
C     ----------------------------------------------------------------------
C     ----------------------------------------------------------------------
      SUBROUTINE MSIP(R,RR,S,SS,SO,T,W,PA,TREF,X,CPL,VA6,TO,ITYPE)
      DOUBLE PRECISION R(99,99),RR(99,99),S(99,99),SS(99,99),SO(99,99)
      DOUBLE PRECISION B(99,99),C(99,99),D(99,99),E(99,99),G(99,99)
      DOUBLE PRECISION V(99,99),H(99,99),W(99,99),PA(99,99),X(99,99)
      DOUBLE PRECISION DEL(99,99),TO(99,99),T(99,99),F(99,99)
      DIMENSION ITYPE(99,99)
      COMMON /SEA4/ALPHA,IS,IFIN,JJ,LLL,JS
      DO 10 J=JJ-1,LLL+1
      DO 10 I=IS-1,IFIN+1
      TO(I,J)=T(I,J)
      B(I,J)=0.
      C(I,J)=0.
      D(I,J)=0.
      E(I,J)=0.
      F(I,J)=0.
      G(I,J)=0.
      H(I,J)=0.
      V(I,J)=0.
   10 DEL(I,J)=0.
      DO 20 J=JJ,LLL
      DO 20 I=IS,IFIN
      IF(ITYPE(I,J).EQ.0)GO TO 20
      B(I,J)=-SS(I,J)/(1.-ALPHA*F(I,J-1)*F(I+1,J-1))
      C(I,J)=-B(I,J)*F(I,J-1)
      D(I,J)=(-RR(I,J)-B(I,J)*G(I,J-1))/
     1(1.+2.*ALPHA*G(I-1,J))
      E(I,J)=PA(I,J)*TREF-B(I,J)*H(I,J-1)-C(I,J)*G(I+1,J-1)-D(I,J)
     1*F(I-1,J)+2.*ALPHA*(C(I,J)*F(I+1,J-1)+D(I,J)*G(I-1,J))
      F(I,J)=(-R(I,J)-C(I,J)*H(I+1,J-1)-2.*ALPHA*
     1C(I,J)*F(I+1,J-1))/E(I,J)
      G(I,J)=-D(I,J)*H(I-1,J)/E(I,J)
      H(I,J)=(-S(I,J)-ALPHA*D(I,J)*G(I-1,J))/E(I,J)
      V(I,J)=(R(I,J)*TO(I+1,J)+RR(I,J)*TO(I-1,J)+S(I,J)
     1*TO(I,J+1)+SS(I,J)*TO(I,J-1)-PA(I,J)*TREF*TO(I,J)+SO(I,J)
     2+X(I,J)-D(I,J)*V(I-1,J)-B(I,J)*V(I,J-1)-C(I,J)*V(I+1,J-1))/E(I,J)
   20 CONTINUE
      DO 30 J=LLL,JJ,-1
      DO 30 I=IFIN,IS,-1
      DEL(I,J)=V(I,J)-F(I,J)*DEL(I+1,J)-H(I,J)*DEL(I,J+1)-G(I,J)
     1*DEL(I-1,J+1)
   30 CONTINUE
      DO 40 J=JJ,LLL
      DO 40 I=IS,IFIN
      T(I,J)=DEL(I,J)+TO(I,J)
   40 W(I,J)=PA(I,J)*T(I,J)*TREF-X(I,J)
      RETURN
      END
C     ------------------------------------------------------------------
C     ------------------------------------------------------------------
C             SUBROUTINE SIP SOLVES AN N*M MATRIX OF EQUATIONS
C             USING THE STRONGLY IMPLICIT PROCEDURE
C     ------------------------------------------------------------------
C     ------------------------------------------------------------------
      SUBROUTINE SIP(R,RR,S,SS,SO,T,H,PA,TREF,X,CPL,VA6,TO,ICOUNT,ITYPE)
      DOUBLE PRECISION R(99,99),RR(99,99),S(99,99),SS(99,99)
      DOUBLE PRECISION SO(99,99),T(99,99),V(99,99)
      DOUBLE PRECISION H(99,99),PA(99,99),X(99,99),TO(99,99),DEL(99,99)
      DIMENSION ITYPE(99,99)
      DOUBLE PRECISION F(99,99),B(99,99),C(99,99),D(99,99),E(99,99)
      COMMON /SEA4/ALPHA,IS,IFIN,JJ,LLL,JS
      DO 10 J=JJ-1,LLL+1
      DO 10 I=IS-1,IFIN+1
      B(I,J)=0.
      C(I,J)=0.
      D(I,J)=0.
      E(I,J)=0.
      F(I,J)=0.
      V(I,J)=0.
      TO(I,J)=T(I,J)
   10 DEL(I,J)=0.
      IF((ICOUNT/2).NE.(.5*ICOUNT))GO TO 40
      DO 20 J=JJ,LLL
      DO 20 I=IS,IFIN
      IF(ITYPE(I,J).EQ.0)GO TO 20
      B(I,J)=-SS(I,J)/(1.+ALPHA*E(I,J-1))
      C(I,J)=-RR(I,J)/(1.+ALPHA*F(I-1,J))
      D(I,J)=PA(I,J)*TREF+ALPHA*(B(I,J)*E(I,J-1)+C(I,J)*F(I-1,J))
     1-B(I,J)*F(I,J-1)-C(I,J)*E(I-1,J)
      E(I,J)=(-R(I,J)-ALPHA*B(I,J)*E(I,J-1))/D(I,J)
      F(I,J)=(-S(I,J)-ALPHA*C(I,J)*F(I-1,J))/D(I,J)
      V(I,J)=(SO(I,J)+X(I,J)+SS(I,J)*TO(I,J-1)+RR(I,J)*
     1TO(I-1,J)-PA(I,J)*TREF*TO(I,J)+R(I,J)*TO(I+1,J)+S(I,J)*TO(I,J+1)
     2-B(I,J)*V(I,J-1)-C(I,J)*V(I-1,J))/D(I,J)
   20 CONTINUE
      DO 30 J=LLL,JJ,-1
      DO 30 I=IFIN,IS,-1
   30 DEL(I,J)=V(I,J)-E(I,J)*DEL(I+1,J)-F(I,J)*DEL(I,J+1)
      GO TO 70
   40 DO 50 J=JJ,LLL,1
      DO 50 I=IFIN,IS,-1
      B(I,J)=-SS(I,J)/(1.+ALPHA*E(I,J-1))
      C(I,J)=-R(I,J)/(1.+ALPHA*F(I+1,J))
      D(I,J)=PA(I,J)*TREF+ALPHA*(B(I,J)*E(I,J-1)+C(I,J)*F(I+1,J))
     1-B(I,J)*F(I,J-1)-C(I,J)*E(I+1,J)
```

```
      E(I,J)=(-RR(I,J)-ALPHA*B(I,J)*E(I,J-1))/D(I,J)
      F(I,J)=(-S(I,J)-ALPHA*C(I,J)*F(I+1,J))/D(I,J)
      V(I,J)=(SO(I,J)+X(I,J)+SS(I,J)*TO(I,J-1)+RR(I,J)*
     1TO(I-1,J)-PA(I,J)*TREF*TO(I,J)+R(I,J)*TO(I+1,J)+S(I,J)*TO(I,J+1)
     2-B(I,J)*V(I,J-1)-C(I,J)*V(I+1,J))/D(I,J)
   50 CONTINUE
      DO 60 J=LLL,JJ,-1
      DO 60 I=IS,IFIN,1
   60 DEL(I,J)=V(I,J)-E(I,J)*DEL(I-1,J)-F(I,J)*DEL(I,J+1)
   70 DO 80 J=JJ,LLL
      DO 80 I=IS,IFIN
      T(I,J)=DEL(I,J)+TO(I,J)
   80 H(I,J)=PA(I,J)*T(I,J)*TREF-X(I,J)
      RETURN
      END
C     ------------------------------------------------------------------
C     ------------------------------------------------------------------
C          SUBROUTINE TDMA SOLVES A TRIDIAGONAL SYSTEM OF EQUATIONS
C     ------------------------------------------------------------------
C     ------------------------------------------------------------------
      SUBROUTINE TDMA(R,RR,S,SS,SO,T,NIP,PA,H,X,TOLD,ITYPE)
      DIMENSION ITYPE(99,99)
      DOUBLE PRECISION A(99,99),B(99,99),C(99,99),D(99,99)
      DOUBLE PRECISION R(99,99),S(99,99),RR(99,99),SS(99,99),SO(99,99)
      DOUBLE PRECISION PA(99,99),TOLD(99,99),X(99,99),H(99,99),T(99,99)
      COMMON /SEA1/HSMP,HLMP,AKS,AKL,YL1,CPS,CPL,ZOTOT1,ZOTOT2,INC,DES
      COMMON /SEA2/IRF,NFSP,IRP,JSH,N,M,TIN,TREF,VA6,TMP,PSS,TR
      COMMON /SEA4/ALPHA,IS,IFIN,JJ,LLL,JS
      IF((.5*NIP).NE.(NIP/2))GO TO 60
      DO 10 J=JJ,LLL
      DO 10 I=IS,IFIN
      TOLD(I,J)=T(I,J)
      D(I,J)=PA(I,J)*TREF
      C(I,J)=SO(I,J)+X(I,J)
      B(I,J)=-SS(I,J)
   10 A(I,J)=-S(I,J)
      JK=JJ+1
      DO 50 I=IS,IFIN
      JFIN=LLL
      DO 20 J=JJ,LLL
      IF((ITYPE(I,J).NE.0).AND.(ITYPE(I,J+1).EQ.0))JFIN=J
   20 CONTINUE
      DO 30 J=JK,JFIN
      Z=B(I,J)/D(I,J-1)
      D(I,J)=D(I,J)-Z*A(I,J-1)
   30 C(I,J)=C(I,J)-Z*C(I,J-1)
      C(I,JFIN)=C(I,JFIN)/D(I,JFIN)
      DO 40 J=JK,JFIN
      K=JFIN-J+JJ
   40 C(I,K)=(C(I,K)-A(I,K)*C(I,K+1))/D(I,K)
      DO 50 J=JJ,JFIN
   50 T(I,J)=C(I,J)
      GO TO 160
   60 DO 70 J=JJ,LLL
```

```
      DO 70 I=IS,IFIN
      TOLD(I,J)=T(I,J)
      D(I,J)=PA(I,J)*TREF
      C(I,J)=SO(I,J)+X(I,J)
      B(I,J)=-RR(I,J)
   70 A(I,J)=-R(I,J)
      DO 150 J=JJ,LLL
      IN=IS
      IX=IS
   80 ISTOP=IFIN
      ISUM=0
      DO 100 I=IN,IFIN
      ISUM=ISUM+ITYPE(I,J)
      IF((ITYPE(I,J).EQ.3).OR.(ITYPE(I,J).EQ.5))IX=I
      IF((ITYPE(I,J).NE.4).AND.(ITYPE(I,J).NE.6))GO TO 90
      ISTOP=I
      GO TO 110
   90 IF((ITYPE(I,J).NE.7).AND.(ITYPE(I,J).NE.8))GO TO 100
      T(I,J)=C(I,J)/D(I,J)
      ISUM=ISUM-ITYPE(I,J)
  100 CONTINUE
  110 IF(ISUM.EQ.0)GO TO 150
      IN=I+1
      JK=IX+1
      DO 120 I=JK,ISTOP
      Z=B(I,J)/D(I-1,J)
      D(I,J)=D(I,J)-Z*A(I-1,J)
  120 C(I,J)=C(I,J)-Z*C(I-1,J)
      C(ISTOP,J)=C(ISTOP,J)/D(ISTOP,J)
      DO 130 I=JK,ISTOP
      K=ISTOP-I+IX
  130 C(K,J)=(C(K,J)-A(K,J)*C(K+1,J))/D(K,J)
      DO 140 I=IX,ISTOP
      IF((ITYPE(I,J).NE.0).AND.(ITYPE(I,J).NE.7).AND.(ITYPE(I,J).NE.8))
     1T(I,J)=C(I,J)
  140 CONTINUE
      IF((IN.LT.IFIN).AND.(IN.NE.IS))GO TO 80
  150 CONTINUE
  160 DO 170 J=INC,LLL
      DO 170 I=IS,IFIN
  170 H(I,J)=(PA(I,J)*T(I,J)*TREF)-X(I,J)
      RETURN
      END
C     ------------------------------------------------------------------
C     ------------------------------------------------------------------
C       SUBROUTINE ICE DETERMINES SHEDDING,REFREEZING AND GROWTH OF ICE
C     ------------------------------------------------------------------
C     ------------------------------------------------------------------
      SUBROUTINE ICE(NIP,T,H,TO,HO,MM,YP,RATE,DTAU,EL,XP,ITYPE,DY,DX,
     1AAK,PA)
      DIMENSION MM(29),IOLD(99,99),ITYPE(99,99)
      DOUBLE PRECISION T(99,99),H(99,99),TO(99,99),HO(99,99),PA(99,99)
      DOUBLE PRECISION YP(99),XP(99),RATE(99),DY(99),AAK(99,99)
      DOUBLE PRECISION EL(29),DX(99)
      COMMON /SEA1/HSMP,HLMP,AKS,AKL,YL1,CPS,CPL,XTOT1,XTOT2,INC,DES
      COMMON /SEA2/IRF,NFSP,IRP,JSH,N,M,TIN,TREF,VA6,TMP,PSS
      COMMON /SEA5/NP,L,JMAX,ISH,KSH,HAFP,NISP
C     ------------------------------------------------------------------
C            DETERMINATION OF WHETHER TEMPERATURES AND ENTHALPYS
C            FOR THE ICE LAYER SHOULD BE STORED FOR THE PURPOSE
C            OF RECREATING ICE AND SHEDDING THE ICE
C     ------------------------------------------------------------------
      IF((IRF.EQ.1).OR.(NIP.LT.NFSP).OR.(IRP.EQ.1).OR.(JSH.EQ.1))GOTO 20
      DO 10 I=2,N
      DO 10 J=INC,M
      T(I,J)=TIN
      H(I,J)=CPL*TREF*(T(I,J)-VA6)
      IF(T(I,J).LE.TMP) H(I,J)=CPS*T(I,J)*TREF
      TO(I,J)=T(I,J)
   10 HO(I,J)=H(I,J)
      IRP=1
      MM(L+1)=M
      IF((JSH.EQ.2).AND.(NIP.LT.NFSP))MM(L+1)=INC
      IF((JSH.EQ.3).AND.(NIP.GE.NFSP))MM(L+1)=INC
C     -------------------------------------------------------
C                VARIABLE ICE GROWTH ALGORITHM
C     -------------------------------------------------------
   20 M=MM(L+1)
      PSS=PSS+1.
      PS=0.
      DO 30 I=1,NP
      YPOLD=YP(I)
      YNEW=300.*RATE(I)*DTAU*PSS
      YP(I)=YP(I)+YNEW
      IF(YPOLD.EQ.YP(I))GO TO 30
      PS=1.
      EL(L)=AMAX1(EL(L),YP(I))
      ELDY=EL(L)/DY(M)
      MM(L+1)=IFIX(ELDY)+INC
   30 CONTINUE
      DO 40 J=1,MM(L+1)
      DO 40 I=2,N
   40 IOLD(I,J)=ITYPE(I,J)
      IF(PS.EQ.0.)GO TO 60
      CALL TYPE(XP,YP,NP,ITYPE,JMAX,DY(M),DX,N,INC)
      MM(L+1)=INC+JMAX
      MM(L+2)=MM(L+1)+1
      PSS=0.
      DO 50 J=1,MM(L+1)
      DO 50 I=2,N
      IF((IOLD(I,J).NE.0).OR.(ITYPE(I,J).EQ.0))GO TO 50
      T(I,J)=T(I,J-1)
      TO(I,J)=TO(I,J-1)
      HO(I,J)=HO(I,J-1)
      DY(J)=DY(J-1)
      AAK(I,J)=AAK(I,J-1)
      PA(I,J)=PA(I,J-1)
   50 IF(ITYPE(I,J).EQ.0)T(I,J)=0.
```

```
C     -----------------------------------------------------
C        DETERMINATION OF WHETHER ICE LAYER SHOULD BE SHED
C     -----------------------------------------------------
   60 IF((ISH.EQ.1).OR.(JSH.EQ.3)) GO TO 70
      IF((JSH.EQ.2).AND.(NIP.LT.NFSP)) GO TO 70
      HINC=H(2,INC+KSH-1)
      HINC=HAFP+(HINC-HSMP)/2.
      IF(HINC.LT.HAFP) GO TO 70
      JSH=2
      IF(IRP.EQ.1) JSH=3
      NISP=NIP
   70 RETURN
      END
/*
//GO.FT06F001 DD SYSOUT=A,OUTLIM=99990
//GO.SYSIN  DD *
-----------------------------------------------------------------------
                    DATA   FOR   THE   PROBLEM
-----------------------------------------------------------------------
       COMMENTS:
        IF PHASE CHANGE IS CONSIDERED IG=2
        IF PHASE CHANGE IS NOT CONSIDERED IG=1
        FOR CONSTANT TEMPERATURE BOUNDARY CONDITIONS AT X=1 IBC1=1
         "      "        "          "        "      AT X=2 IBC2=1
         "  CONVECTION BOUNDARY CONDITIONS          AT X=1 IBC1=2
         "      "          "        "               AT X=2 IBC2=2
        IF NO HEAT SOURCE IS PRESENT                IH  = 1
        IF POINT HEAT SOURCE IS PRESENT             IH  = 2
        IF INTERNAL HEAT GENERATION IN A SLAB OCCURS IH  = 3
        IF CONSTANT HEAT INPUT IS USED              IHQ = 1
        IF STEP FUNCTION HEAT INPUT IS USED         IHQ = 2
        IF RAMP FUNCTION HEAT INPUT IS USED         IHQ = 3
        IF SINE FUNCTION HEAT INPUT IS USED         IHQ = 4
-----------------------------------------------------------------------
DATA FOR THE COMPOSITE BODY
***************************
TOTAL NUMBER OF SLABS IN THE Y-DIRECTION              L= 006
TOTAL NUMBER OF LAYERS IN THE X-DIRECTION            NX= 003
DATA FOR EACH SLAB:
----------------
NUMBER OF NODES    LENGTH      THERMAL CONDUCTIVITY  THERMAL DIFFUSIVITY
                  IN INCHS.    IN B.T.U./HR..FT.'F.     IN FT*FT/HR
       07        0000.08700      000066.500000       000001.650000
       12        0000.05000      000000.220000       000000.008700
       03        0000.00400      000007.600000       000000.138000
       05        0000.01000      000000.220000       000000.008700
       05        0000.01200      000008.700000       000000.150000
       21        0000.25000      000001.290000       000000.044600

 NUMBER OF NODES   LENGTH      THERMAL CONDUCTIVITY  THERMAL DIFFUSIVITY
 IN X-DIRECTION   IN INCHS.       IN B.T.U./HR..FT.'F.    IN FT*FT/H
       03        0000.12500      000000.220000       000000.008700
       06        0000.25000      000007.600000       000000.138000
       03        0000.12500      000000.220000       000000.008700
```

```
--------------------
DATA FOR THE HEATER:
--------------------
TYPE OF HEAT SOURCE                                     IH= 003
FOR IH=2  POINT HEAT SOURCE       BETWEEN SLAB         L1= 002
                                      AND  SLAB         L2= 003
FOR IH=3  INTERNAL HEAT GENERATION IN SLAB             IJ= 003
TYPE OF HEAT SOURCE USED                               IHQ= 001
CONSTANT HEATER  ON-TIME  OFF-TIME     VARIABLE HEATER    LAG TIME
IN WATTS/IN**2                         IN WATTS/IN**2
  00000.0000      010.00    050.00       00000.0000         000.00
  00030.0000      010.00    050.00       00030.0000         000.00
  00000.0000      010.00    050.00       00000.0000         000.00
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
BOUNDARY AND INITIAL CONDITIONS:
-----------------------------
TYPE OF BOUNDARY CONDITION                AT J=1  IBC1= 002
                                          AT J=M  IBC2= 002
                                          AT I=1  IBC3= 002
                                          AT I=N  IBC4= 002
   IBC1=1, CONSTANT TEMPERATURE  AT J=1  TX1 = 000010.000000 DEG.F
   IBC2=1,     "          "      AT J=M  TX2 = 000010.000000 DEG.F
   IBC3=1, CONSTANT TEMPERATURE  AT I=1  TX3 = 000010.000000 DEG.F
   IBC4=1,     "          "      AT I=N  TX4 = 000010.000000 DEG.F

   IBC1=2, AMBIENT       "       AT J=1  TG1 = 000010.000000 DEG.F
           HEAT TRANSFER COEFF   AT J=1   H1 = 000001.000000 BTU/H.F.FT2
   IBC2=2, AMBIENT  TEMPERATURE  AT J=M  TG2 = 000010.000000 DEG.F
   IBC3=2, AMBIENT       "       AT I=1  TG3 = 000010.000000 DEG.F
           HEAT TRANSFER COEFF   AT I=1   H3 = 000000.000000 BTU/H.F.FT2
   IBC4=2, AMBIENT  TEMPERATURE  AT I=N  TG4 = 000010.000000 DEG.F
           HEAT TRANSFER COEFF.  AT I=N   H4 = 000000.000000 BTU/H.F.FT2
THE INITIAL TEMPERATURE IN THE SLAB      TIN = 000010.000000 DEG.F
THE REFERENCE TEMPERATURE               TREF = 000032.000000 DEG.F

             OUTER AMBIENT HEAT TRANSFER COEFFICIENTS
    000150.000    000150.000    000150.000    000150.000    000150.000
    000150.000    000150.000    000150.000    000150.000    000150.000
    000150.000    000150.000    000150.000    000150.000    000150.000
---------------------------------------------------------------------
PROPERTIES OF WATER:

   THE PHASE CHANGE CONSIDERED IF YES IG=2 IF NO IG=1    IG= 002
LATENT HEAT OF ICE                    HLAM = 000143.400000 B.T.U./LB
THERMAL DIFFUSIVITY OF WATER          ALPL = 000000.005100 FT*FT/HR.
   "    CONDUCTIVITY "   "            AKL  = 000000.320000 B.T.U./HR.FT.F
DENSITY OF WATER                      DEL  = 000062.400000 LB./CU.FT
SPECIFIC HEAT*DENSITY OF WATER        CPL  = 000062.212800 B.T.U./CU.FT
DENSITY OF ICE                        DES  = 000057.400000 LB./CU.FT
SPECIFIC HEAT*DENSITY OF ICE          CPS  = 000028.814800 B.T.U./CU.FT
THE MELTING TEMPERATURE               TMP  = 000032.000000 DEG.'F.
NUMBER OF POINTS FOR ICE SHAPE APPROXIMATION  NP= 002
X-VALUE   0000.00000  Y-VALUE 0000.25000    RATE  0000.00000
          0000.50000          0000.25000          0000.00000
```

```
.........................................................................................
  ADDDITIONAL DATA:
  ::::::::::::::::::::
 INITIAL TIME STEP                               DTAUI = 000000.100000 SECS.
                    FOR TIME STEPS               NISP  = 0100
 INTERMIDIATE TIME STEP                          DTAUM = 000000.100000 SECS.
                    FOR TIME STEPS               NMSP  = 0100
 FINAL TIME STEP                                 DTAUF = 000000.100000 SECS.

  PRINT ENTHALPYS IN ICE; KENTH=2                KENTH = 001
 THE SHEDDING ICE LAYER CONSIDERED IF YES ISH=2 IF NO ISH=1   ISH=   01
 DOES THE INITIAL TEMPERATURE CHANGE IF YES IRF=2 IF NO IRF=1   IRF= 01
 REFREEZING OF ICE LAYER OCCURS AT TIME STEP           NFSP = 000
  SHEDDING OCCURS AT NODE                        KSH   = 001
  FOR SHEDDING AT 2ND CYCLE FOR TIME STEPS       NHSP  = 000

 FREQUENCY OF TIME STEP PER PRINT OF OUTPUT IFREQ= 00010
  FOR TIME STEPS STOP THE PROGRAM                NTSP  = 000200
  MAXIMUM ITERATIONS PER TIME STEP             JCOUNT  = 0999
  100*[T-TOLD]/T IS LESS THAN                 EPSI   = 0.00100
 INITIAL ACELLERATION PARAMETER(SOR)           WI      = 000001.700000
                    FOR TIME STEPS             NIN     = 170
 INTERMIDIATE ACELLERATION PARAMETER(SOR)      WM      = 000001.500000
                    FOR TIME STEPS             NMED    = 200
 FINAL ACCELLERATION PARAMETER(SOR)            WF      = 000001.300000

  'METHOD' DETERMINES THE NUMERICAL METHOD USED IN THE PROGRAM
   IF = 1, USE COMBINED METHOD(EXPLICIT,CRANK-NICOLSON,IMPLICIT)
   IF = 2, USE HOPSCOTCH METHOD
   IF = 3, USE ADE(ALTERNATING DIRECTION EXPLICIT) METHOD
   IF = 4, USE ADI(ALTERNATING DIRECTION IMPLICIT) METHOD
   IF = 5, USE TIME-SPLITTING METHOD
   IF = 6, USE SIP(STRONGLY IMPLICIT PROCEDURE)
   IF = 7, USE MSIP(MODIFIED STRONGLY IMPLICIT PROCEDURE)

  FINITE DIFFERENCING PROCEDURE               METHOD = 04
  VALUE OF THETA FOR COMBINED METHOD          THETA  = 0.50000

  ALPHA IS A DAMPING FACTOR FOR THE MSIP PROCEDURE
  VALUE FOR ALPHA ( 0.3 < ALPHA < 0.6 )                = 0.50000
/*
//
```

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No. NASA CR-4202	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle
A Comparison of Numerical Methods for the Prediction of Two-Dimensional Heat Transfer in an Electrothermal Deicer Pad

5. Report Date
December 1988

6. Performing Organization Code

7. Author(s)
William B. Wright

8. Performing Organization Report No.
None (E-4380)

10. Work Unit No.
505-68-11

9. Performing Organization Name and Address
University of Toledo
Department of Chemical Engineering
Toledo, Ohio 43606

11. Contract or Grant No.
NAG3-72

13. Type of Report and Period Covered
Contractor Report
Final

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135-3191

14. Sponsoring Agency Code

15. Supplementary Notes

Project Manager, Mario Vargas, Propulsion Systems Division, NASA Lewis Research Center. This report was a thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Chemical Engineering to The University of Toledo in February 1988.

16. Abstract

Transient, numerical simulations of the deicing of composite aircraft components by electrothermal heating have been performed in a two-dimensional rectangular geometry. Seven numerical schemes and four solution methods were used to find the most efficient numerical procedure for this problem. The phase change in the ice was simulated using the Enthalpy method along with the Method of Assumed States. Numerical solutions illustrating deicer performance for various conditions are presented. Comparisons are made with previous numerical models and with experimental data. The simulation can also be used to solve a variety of other heat conduction problems involving composite bodies.

17. Key Words (Suggested by Author(s))
Deicer

18. Distribution Statement
Unclassified – Unlimited
Subject Category 06

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No of pages 140	22. Price* A07

NASA FORM 1626 OCT 86

*For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1988